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, 53706 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. Under stochastic simulation of a genomic selection program, COMA maintained a target inbreeding rate of 0.5% using either pedigree or genomic IBD kinship. Significantly more genetic gain was realized with pedigree kinship, which is consistent with previous studies showing the selective advantage of an individual under OCS is dominated by its Mendelian sampling term. Despite the higher accuracy (+0.2–0.3) when predicting mate performance with OMA compared to OCS, there was little long-term gain advantage. The sparsity of the COMA mating design and flexibility to incorporate mating constraints offer practical incentives over OCS. In a potato breeding case study with 170 candidates, the optimal solution at 0.5% inbreeding involved 43 parents but only 43 of the 903 possible matings.

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 when plant varieties are released as inbred lines or F1 hybrids (e.g., wheat, maize), long-term management of genetic diversity by inbreeding control is still relevant with respect to an outbred parental population, which may be defined theoretically or used in practice (Gaynor et al. 2017).

Because individuals with high genetic merit are often related, 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 and is inversely related to the effective population size (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). There is also no published formula for the genomic prediction of mate performance (GPMP) in polyploids, defined as the F1 progeny mean with directional 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 the *convex* nature of the 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 evolutionary optimization algorithms (Holland 1992). 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 into the population mean plus additive and digenic dominance values (Endelman 2023). (Higher order dominance for polyploids is neglected.) The additive (u_i) and dominance (v_i) values for individual i are the products of regressors and regression coefficients:

$$u_i = W_i\alpha = \phi[p_i - p]\alpha$$
$$v_i = Q_i\beta = [\tau - \phi^2(p_i - \xi)^2]\beta \qquad [1]$$

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

Additive regressor W_i is a linear function of the difference between the individual allele frequency p_i and the population allele frequency $p = \overline{p}_{\cdot} = 1 - q$. 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 Hardy-Weinberg (i.e., panmictic) Equilibrium, HWE (Falconer and Mackay 1996).

Dominance regressor Q_i is a quadratic function of the difference between p_i and ξ, a parameter that depends on p and ϕ. 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 is used to emphasize it is the extension of α to digenic effects.

Consider F1 progeny of the mating between individuals 1 and 2 with allele frequencies p_1 and p_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

$$\text{MPD} = \tfrac{1}{2}(v_1 + v_2) = \beta\left[\tfrac{-\phi}{2}(p_1^2 + p_2^2) + \xi\phi^2(p_1 + p_2) - \phi(\phi - 1)p^2\right] \qquad [2]$$

This expression 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 $(p_1\,,\,p_2)$ plane. The general form of the quadratic equation for conic sections is typically written as

$$Ap_1^2 + Bp_1p_2 + Cp_2^2 + Dp_1 + Ep_2 + F \qquad [3]$$

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

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

$$\text{where } \mathbf{M} = \begin{bmatrix} A & B/2 \\ B/2 & A \end{bmatrix} \text{ and } \boldsymbol{\theta} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\left(\tfrac{D}{2A+B}\right) \qquad [4]$$

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 under a random bivalents model of meiosis, which is mathematically equivalent to random chromosome segregation (Gallais 2003). Gametic dosage γ follows the hypergeometric distribution, which describes the probability of choosing γ copies of the allele in $\phi/2$ draws without replacement from a parent with ϕ chromosomes and zygote dosage *z*. The first and second moments of this distribution are (Ibe 2014)

$$E[\gamma] = \frac{z}{2}$$
$$E[\gamma^2] = \frac{\phi z + (\phi - 2)z^2}{4(\phi - 1)} \qquad [5]$$

Allele dosage z_{12} of the F1 progeny from parents 1 and 2 is the sum of random variables γ_1 and γ_2, and the moments follow from Eq. 5:

$$E[z_{12}] = \frac{1}{2}(z_1 + z_2)$$
$$E[z_{12}^2] = \frac{\phi(z_1 + z_2) + (\phi - 2)(z_1^2 + z_2^2) + 2(\phi - 1)z_1 z_2}{4(\phi - 1)} \qquad [6]$$

From Eq. 1, the expected dominance value of the F1 progeny equals $E[Q_{12}]\beta$, where

$$E[Q_{12}] = \tau - E[(z_{12} - \phi\xi)^2] = (\tau - \phi^2\xi^2) - E[z_{12}^2] + E[z_{12}][2p(\phi - 1) + 1] \qquad [7]$$

The expression $\tau - \phi^2\xi^2$ simplifies to conic parameter *F* in Table 1. The other conic parameters follow by substituting Eq. 6 into Eq. 7 and recognizing that $z_i = \phi p_i$.

The solution for MPH follows by subtracting the conic parameters for MPD from the conic parameters for the F1 mean. 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) (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 $\sqrt{2\phi - 1}$ (derivation in File S1).

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 F1 progeny mean:

OMA: $\max_{\mathbf{x}} \mathbf{g}'\mathbf{x} = \sum_i \sum_j g_{ij} x_{ij}$

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. 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:

$$y_{f,i} = \sum_j x_{ij} \qquad y_{m,j} = \sum_i x_{ij}$$
$$y_i = \frac{1}{2}(y_{f,i} + y_{m,i}) \qquad 1 = \sum_i y_i = \sum_i y_{f,i} = \sum_i y_{m,i} \qquad [8]$$

In the previous section, the F1 progeny mean was expressed as MPA+MPD+MPH. Alternatively, it can be expressed as the sum of general and specific combining abilities (GCA and SCA, respectively): $\mathrm{g}_{ij} = \mathrm{GCA}_i + \mathrm{GCA}_j + \mathrm{SCA}_{ij}$. 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 proportion $\left(\frac{\frac{\phi}{2}-1}{\phi-1}\right)$ of the dominance value (Gallais 2003). If SCA is neglected, the OMA objective reduces to maximizing the average breeding value of the parents, weighted by their contributions (derivation in File S1), which is the typical objective for OCS:

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

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 (Falconer and Mackay 1996). 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. Under random chromosome segregation, 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:

$$\begin{aligned}(\phi - 1)\bar{F}_{t+1} &= (\phi - 1)\textstyle\sum_i \sum_j x_{ij}F_{ij} = \tfrac{\phi}{2}\sum_i \sum_j x_{ij}K_{ij} + \left(\tfrac{\phi}{2} - 1\right)\sum_i y_i F_i \\ &= \tfrac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\tfrac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F}\end{aligned} \qquad [14]$$

The second equality in Eq. 14 follows from the relationship between mate allocations and individual contributions (Eq. 8), 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

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

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

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

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

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

$F_{ij,ab}$ 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*:

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

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):

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

METHODS

Convex Optimization

The convex OMA problem is

$$\begin{aligned}
&\max_{\mathbf{x},\mathbf{y}} \ \mathbf{g}'\mathbf{x} \\
&\mathbf{0} \le \mathbf{L}_\mathrm{x} \le \mathbf{x} \le \mathbf{U}_\mathrm{x} \le \mathbf{1} \\
&\mathbf{0} \le \mathbf{L}_\mathrm{y} \le \mathbf{y} \le \mathbf{U}_\mathrm{y} \le \mathbf{1} \\
&\mathbf{1}'\mathbf{x} = 1 \\
&\mathbf{y} = \mathbf{T}\mathbf{x} \\
&\mathbf{s}'\mathbf{y} = \frac{1}{2} \\
&\mathbf{a} \le \mathbf{C}\mathbf{y} \le \mathbf{b} \\
&F_{t+1,\Delta F_\mathrm{min}} \le \frac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\frac{\phi}{2}-1\right)\mathbf{y}'\mathbf{F} \le F_{t+1,\Delta F_\mathrm{max}} \\
&\frac{\phi}{2}(\phi-1)\mathbf{y}'\mathbf{K}\mathbf{y} + \left(\frac{\phi}{2}-1\right)\left[\frac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\frac{\phi}{2}-1\right)\mathbf{y}'\mathbf{F}\right] \le F_{t+2,\Delta F_\mathrm{max}}
\end{aligned}$$

The genetic merit **g** for each mating was the sum of MPA, MPD, and MPH across all markers. The first two constraints represent lower (**L**) and upper (**U**) bounds on the mate allocation (**x**) and

contribution ($\mathbf{y}$) variables. The incidence matrix $\mathbf{T}$ in constraint $\mathbf{y} = \mathbf{Tx}$ captures the relationship between the variables in Eq. 8. 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 $\mathbf{a} \leq \mathbf{Cy} \leq \mathbf{b}$ represents optional linear constraints on the contributions; an example is given for the potato dataset. The constraint involving F_{t+1} combines Eq. 10 and 16, putting lower and upper bounds on the inbreeding coefficient of the progeny based on lower and upper bounds on the inbreeding rate: $\Delta F_{\min} \leq \Delta F \leq \Delta F_{\max}$. The last constraint combines Eq. 11 and 19, putting an upper bound on the inbreeding coefficient of the grand-progeny; $\mathbf{K}$ must be positive semidefinite for convexity.

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

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

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

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

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

$$\mathbf{a} \leq \mathbf{Cy} \leq \mathbf{b}$$

$$\frac{\phi}{2}\mathbf{y}'\mathbf{Ky} + \left(\frac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F} \leq F_{t+1,\Delta F_{\max}}$$

The most appropriate inbreeding constraint, Eq. 15, is not convex, so the hermaphroditic form, Eq. 16, is used instead. This is the traditional OCS approach (Woolliams et al. 2015), but as Eq. 19 illustrates, it is a tighter bound than strictly needed for diploid species with separate sexes.

R package COMA (https://github.com/jendelman/COMA) uses R package CVXR (Fu et al. 2020) to solve convex formulations of OCS and OMA (File S3). CVXR transforms high-level 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, 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 (MOSEK ApS 2024).

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

Marker effects were computed from the 2015–2019 data of the R/StageWise package (Endelman 2023), which contained three traits—total yield, fry color, and vine maturity—and 12K SNP markers for 700 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 (Butler et al. 2023) to estimate variance-covariance (var-cov) parameters. 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 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 for clones vs. traits:

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

The trait var-cov matrices ($\mathbf{\Sigma}$) were unstructured (i.e., full rank). The additive **G** and dominance **D** genomic relationship matrices follow the multi-locus extension of Eq. 1, using allele frequency p_k at locus k for the current population (Endelman 2023):

$$\begin{aligned} \mathbf{G} &= \frac{\mathbf{W}\mathbf{W}^{\mathrm{T}}}{\phi \sum_k p_k q_k} \\ \mathbf{D} &= \frac{\mathbf{Q}\mathbf{Q}^{\mathrm{T}}}{\binom{\phi}{2} \sum_k 4 p_k^2 q_k^2} \\ \mathbf{f}_{\mathrm{D}} &= \frac{-\mathbf{Q}\mathbf{1}}{\binom{\phi}{2} \sum_k 2 p_k q_k} \end{aligned} \qquad [21]$$

The **W** and **Q** matrices contain the additive and dominance regressors, respectively (Eq. 1). The elements of covariate $\mathbf{f}_D$ are genomic inbreeding coefficients, and the corresponding vector of regression coefficients **b** (Eq. 20), one for each trait, represents inbreeding depression, aka baseline heterosis (Labroo et al. 2021).

Empirical BLUPs for marker effects were based on a multi-trait index using R/StageWise, with the contribution of inbreeding depression included in the dominance effects (Endelman 2023). Yield and fry color have similar importance as breeding objectives, while later maturity is undesirable. Initially, a restricted index was calculated, with equal merit for yield and fry color and zero gain for maturity (File S5). Due to the unfavorable correlation between yield and late maturity, the yield response for this index was only 65% of the response for fry color. As an alternative, a desired gains index was calculated (Pesek and Baker 1969), based on equal gains for yield and fry color and zero gain for maturity. The restricted index methodology was described in Endelman (2023); from Eq. 24 of that publication, if $\boldsymbol{\delta}$ denotes the desired gain vector, then the index coefficients of the multi-trait BLUPs are $\mathbf{c} = \mathbf{B}^{-1}\boldsymbol{\delta}$, where $\mathbf{B}^{-1}$ is the inverse var-cov matrix of the BLUPs. The realized response is proportional to the desired gains, and from Eq. 25 of Endelman (2023), the proportionality constant is $i/\sqrt{\boldsymbol{\delta}'\mathbf{B}^{-1}\boldsymbol{\delta}}$ for selection intensity i. Imposing equal gains for yield and fry color only reduced total merit by 1% (File S5), and given the uncertainty of the true merit coefficients, this index was selected.

Estimates of dominance variance and inbreeding depression for the multi-trait index were expressed relative to the additive variance and standard deviation, respectively (File S5, Table S1 in File S2). If **v** denotes the multi-trait vector of dominance values (Eq. 20), the vector of dominance index values is $(\mathbf{I}\otimes\mathbf{c}')\mathbf{v}$, with mean $\boldsymbol{\mu} = -(\mathbf{c}'\mathbf{b})\mathbf{f}_D$ and variance $\mathbf{V} = (\mathbf{c}'\boldsymbol{\Sigma}_D\mathbf{c})\mathbf{D}$. The dominance variance of the population is therefore $\left[\overline{\mathrm{diag}(\mathbf{V})} - \overline{\mathrm{V}}_{..}\right] + \left[\overline{\mu_{.}^2} - (\overline{\mu}_{.})^2\right]$ (Legarra 2016; Endelman 2023), and the additive variance was computed analogously.

The PVY resistance data are based on KASP marker snpST00073 for Ry_{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 was simulated with AlphaSimR (Gaynor et al. 2021), using a constant population size of 500 hermaphroditic individuals and non-overlapping generations. R/SimPlus (https://github.com/jendelman/SimPlus) was created to standardize breeding and analysis modules across the treatments (File S4).

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.

AlphaSimR uses the (a,d) parameterization for QTL effects, where $2a$ is the difference between the homozygotes and $a+d$ is the difference between the heterozygote and recessive homozygote (Falconer and Mackay 1996). An "ADG" trait was simulated with the following settings: nQtlPerChr=1000, mean=0, var=1, varDD=0.5, meanDD=0.5, useVarA=FALSE. This implies a normal distribution for the dominance degree (DD) parameter, $d/|a|$, with mean 0.5, variance 0.5, and effects scaled to create unit genetic variance in the founders. The average ratio of dominance to additive variance in the founder population was 0.12 for diploids and 0.25 for tetraploids, compared to 0.38 for the potato dataset (Table S1 in File S2). The average baseline heterosis, scaled by the additive genetic standard deviation, was 4.3 and 7.3 for the diploid and tetraploid simulations, respectively, compared to 7.1 for the potato dataset. Non-genetic variance (parameter varE) was fixed at 2 throughout the simulation.

10 generations of phenotypic selection were simulated using OCS at 0.5% inbreeding (pedigree kinship), followed by 25 generations of genomic selection using different methods. (Generation 0 designates the beginning of GS.) Each GS cycle consisted of (1) phenotyping the candidates, (2) calculating marker effects, (3) developing the mating plan by OCS or OMA, (4) and generating new progeny. To reduce computing time, the training population for predicting marker effects was limited to phenotype and genotype data for the selection candidates and two previous generations. 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. 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.

Due to the large number of mate allocation variables (125,250) for 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 upper bound was increased in increments of 0.5% (software parameter *dF.step*).

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_{IBD} is genomic kinship and extends method A to capture Mendelian sampling; it is equivalent to an allele matching coefficient using founder alleles, which were obtained in AlphaSimR with *pullIbdHaplo*. Method G is Method 1 of VanRaden (2008), using founder allele frequencies as the reference population (Eq. 21). Computation of the $\mathbf{G}_{IBD}$ and $\mathbf{G}$ matrices was implemented as function *G_mat* in R/polyBreedR (Endelman et al. 2024). Inbreeding coefficients were computed from the diagonals of the kinship matrix according to $f_{A,i} = (\phi K_{ii} - 1)/(\phi - 1)$ (Gallais 2003).

The mating plan from COMA was realized in AlphaSimR using *SimPlus::sim_mate*, which generates progeny according to the vector of mate allocations $\mathbf{x}$, interpreted as a discrete probability distribution. The mating plan under OCS was a diallel with mate allocation equal to the product of the parental contributions.

RESULTS

Constraints and kinship

COMA limits inbreeding in the progeny under the optimized mating plan, and in the grand-progeny assuming random mating of the progeny. The grand-progeny constraint is needed to maintain the target inbreeding rate; without it, the optimal solution would be 100% allocation to the mating with the highest merit for unrelated diploid parents. The grand-progeny constraint is quadratic and therefore, to remain convex, is only an upper bound (as with OCS). The progeny constraint is a linear inequality, which allows for both upper and lower bounds on the inbreeding

rate. Initially, only an upper bound was used, based on the hypothesis that if the solution with maximum genetic merit involved less inbreeding, this would be advantageous for long-term gain.

But this hypothesis was contradicted in a simulated genomic selection program. Comparing COMA selection with upper bound (UB) vs. equality (EQ) constraints at 0.5% inbreeding, genetic gain with the UB strategy was slightly higher after one generation (Fig. 3A, Fig. S2 in File S2). This is expected from the principles of mathematical optimization: since all possible mate allocations under EQ are also allowed under UB, the optimal UB solution cannot be lower and will typically be higher. But when COMA was applied in generation 1, there was more gain with EQ than UB in generation 2; this does not violate any mathematical principle because the candidates are different. When inbreeding was measured by pedigree IBD kinship (A), the long-term gain advantage of EQ over UB was small but statistically significant at the 0.05 level (Fig. 3A, Fig. S2 in File S2). For genomic IBD kinship (G_{IBD}), the rate of genetic gain with EQ was much higher.

The trends for inbreeding coefficient and genetic variance help explain the results for genetic gain. Under the EQ constraint, the target inbreeding rate of 0.5% was achieved for both pedigree and genomic kinship (Fig. 4). But with only an UB constraint, the inbreeding coefficient decreased in the first generation. With pedigree kinship, the inbreeding rate returned to the target value within a few generations, but with genomic kinship, the rate only rebounded to ~0.1%. When partitioning dominance genetic variance into genic variance (which depends only on allele frequencies) and disequilibrium covariance (DCV), the former was unchanged in the first generation, but the latter decreased substantially (Fig. 3B, Table 2). This was observed for both EQ and UB constraints, but whereas some DCV remained under EQ, it was exhausted under UB.

When the two kinship methods were compared under EQ constraints, pedigree kinship resulted in significantly higher gain for both the diploid (Fig. 3A, Fig. S2 in File S2) and tetraploid (Fig. S3 in File S2) simulations.

Comparison with OCS

The COMA software also contains an implementation of OCS, and the two methods were compared at 0.5% and 1.0% inbreeding using pedigree kinship. Both selection methods achieved the target inbreeding rates at both ploidy levels (Fig. S4 in File S2). The number of selected

parents was not significantly different between OMA and OCS (Fig. 5) and was higher than the effective population size, $N_e = \frac{1}{\phi(\Delta F)}$, for ploidy ϕ. At 0.5% inbreeding, 111 diploid and 74 tetraploid parents were selected on average. At 1.0% inbreeding, those numbers decreased to 72 and 36 parents, respectively. The number of matings with OMA was much lower than with OCS, which involved hundreds of random matings based on the optimum contributions. At 1.0% inbreeding, the average number of matings with OMA was only 60 for diploids and 11 for tetraploids.

OMA achieved the same or higher gain as OCS and was significantly more efficient at converting genic variance into gain, but the difference was small (Fig. 6). At 0.5% inbreeding, the gain advantage of OMA persisted for all 25 generations in tetraploids; for diploids, the gain advantage disappeared by 25 generations, but the genic variance with OMA was significantly higher. Similar results were observed at 1.0% inbreeding (Fig. S5 in File S2).

Because OMA includes specific combining ability (SCA), the accuracy for genomic prediction of mate performance (GPMP) was consistently 0.2–0.3 higher than with OCS (Fig. 7). GPMP accuracy was not significantly different between the 2x and 4x simulations with OMA, but with OCS the 4x accuracy was higher (Fig. 7). The difference can be attributed to a smaller SCA variance in the 4x simulation, which occurred because (1) the SCA variance at HWE is $V_D/4$ for 2x but $V_D/6$ for 4x (Gallais 2003); and (2) there was only half as much as V_D (relative to V_A) in the 4x founders (Table S1 in File S2).

Potato example

The COMA tutorial uses data from the University of Wisconsin potato breeding program (File S5). The parental candidates are 170 autotetraploid clones, with 12K SNP array markers and a pedigree kinship matrix. The inbreeding coefficients ranged from 0.00 to 0.09 (median 0.02), and 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 and training population of 700 clones.

As mentioned above, COMA can impose lower (LB) and upper bounds (UB) on the inbreeding rate for the next generation, ΔF_1. The UB is also applied to the average inbreeding rate over two generations, ΔF_2, assuming random mating of the progeny. There was no feasible solution with an UB of 0.5%, which reflects the relatedness of the candidates. The smallest

feasible UB was 0.8% when no LB was imposed, resulting in $\Delta F_1 = -0.8\%$ and a predicted progeny mean of $1.22\sigma_A$. The tendency toward outbreeding is reminiscent of the simulation results, which raises the concern that the population could get trapped in a local optimum. With a LB of 0.5% for ΔF_1, the smallest feasible value for ΔF_2 was 1.0%, and the predicted merit was $1.23\sigma_A$. As seen in the simulation, COMA solutions are sparse: of the 903 possible matings (excluding selfs and reciprocals) for the 43 selected parents, only 43 matings were needed to achieve optimal gain.

The lowest feasible inbreeding rate with OCS was 1.5%. Although the software reports the predicted merit of the optimum solution as $1.22\sigma_A$, this neglects the contribution of SCA. When SCA was included, the predicted merit reduced to $1.04\sigma_A$, which is lower than the OMA solution despite the higher inbreeding rate.

The COMA tutorial shows how additional constraints can be incorporated. Besides the three quantitative traits in the selection index (yield, fry color, and maturity), resistance to potato virus Y (PVY) is an important qualitative trait, which was scored with a genetic marker. For the original COMA solution with $(\Delta F_1, \Delta F_2) = (0.5\%, 1.0\%)$, 19 of the 43 matings had a resistant parent, with an expected allele frequency of 8.6% in the next generation assuming single-copy dosage in the parents (Table 3). One selection strategy would be to require a resistant parent in every mating, which is achieved 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 a reduction in predicted merit from $1.23\sigma_A$ to $1.02\sigma_A$. Alternatively, the allele frequency can be constrained directly with an inequality of the form $\mathbf{s}'\mathbf{y} \geq p$, where $\mathbf{s}$ is the vector of allele frequencies in the parents (i.e., ¼ for single-copy resistant and 0 for susceptible) and p is the target allele frequency. Using this approach to achieve 12.5% R gene frequency, the predicted merit was slightly higher, at $1.07\sigma_A$.

DISCUSSION

Two types of inbreeding coefficient were used in this study. The first, denoted f_D, was introduced as a covariate in the genomic prediction model to account for inbreeding depression. 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 (**G**), although the individual values were not identical. The derivation of this mathematical equality is given in File S1:

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

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

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

Eq. 22, which uses allele frequency p_k at locus k for the current population, indicates $\bar{f} = 0$ when the population variance of the allele dosages, $\mathrm{var}(z)$, equals the variance (ϕpq) of a binomial random variable with the same allele frequency. The maximum value of $\bar{f}$ occurs when there are no heterozygotes, in which case $\mathrm{var}(z) = \phi^2 pq$, so $\bar{f}_{\max} = 1$. The minimum value of $\bar{f}$ occurs when $\mathrm{var}(z) = 0$, so $\bar{f}_{\min} = \frac{-1}{\phi-1}$. The limits of $\bar{f}$ for diploids are [-1,1], consistent with its interpretation as a gametic correlation (Wright 1921; Powell et al. 2010; Endelman and Jannink 2012).

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 (G_{IBD}) kinship were effective at meeting the target inbreeding rate in the simulation. But genetic gain was higher using pedigree kinship (Fig. 3), and similar results with OCS have been reported (Meuwissen et al. 2020). This finding is consistent with theoretical and computational work showing the selective advantage of an individual under OCS is dominated by its Mendelian sampling term (Grundy et al. 1998; Avendaño et al. 2004). Because individuals with selective advantage from the same family typically have higher genomic kinship than the expected value from pedigree, there is less flexibility to increase gain by selecting multiple full-sibs (or half-sibs) under genomic control of inbreeding (Henryon et al. 2019).

What about using **G**, rather than $\mathbf{G}_{IBD}$, for genomic control of inbreeding? This is practically appealing because the latter is computationally intensive for diploids (Meuwissen and Goddard 2010; Whalen et al. 2018) and has not even been reported in autotetraploids; only one generation from multiple founders has been published (Zheng et al. 2021). (In the present study, tetraploid founder haplotypes were tracked during the meiosis simulation, not inferred.) But using **G** for

inbreeding control is not recommended within the current framework. The constraint equations are based on the law of total probability (Eq. 10), while the inbreeding coefficient from **G** is a correlation (Eq. 22), not a probability. De Beukelaer et al. (2017) also highlighted this theoretical discrepancy and verified it by simulation, using the current population as the reference population. It is more common to use reference allele frequencies from the founder population, but this also failed under simulated mass selection with COMA: at a target inbreeding rate of 0.5%, the realized rate exceeded 5%, and genic variance was depleted too rapidly (Fig. S6 in File S2).

My initial hypothesis was that OMA would achieve higher gain than OCS because of its ability to predict SCA. The simulation confirmed that GPMP accuracy was 0.2–0.3 higher (Fig. 7), but the gain advantage was small and did not increase over time (Fig. 6). By contrast, when Werner et al. (2023) compared truncation selection on GEBV vs. GPMP, the latter achieved more than 3 times as much genetic gain after 20 cycles. Inbreeding was not controlled in that study, and the realized inbreeding rate under GEBV selection was ~5% compared to ~1% for GPMP, but this does not appear to explain the dichotomy. Differences in the structure and parameters of the simulated breeding program may be responsible, which indicates more testing is needed.

Even without a gain advantage for COMA, the sparsity of the mating design and flexibility to incorporate mating constraints offer practical incentives over OCS. Earlier versions of COMA have been used for several years in the University of Wisconsin potato breeding program, but the optimal plan was not realized due to frequent and unpredicted problems with reproductive fertility. COMA is used a second time after the crossing is done, converting the realized seed quantities into upper limits for mate allocation based on our greenhouse seedling capacity; the optimal solution guides how many seeds to sow for the next generation.

Acknowledgments

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

References

Akdemir D, Sánchez JI. 2016. Efficient breeding by genomic mating. Front Genet. 7(NOV):210. doi:10.3389/fgene.2016.00210.

Amadeu RR, Cellon C, Olmstead JW, Garcia AAF, Resende MFR, Muñoz PR. 2016. AGHmatrix: R package to construct relationship matrices for autotetraploid and diploid species: a blueberry example. Plant Genome. 9(3). doi:10.3835/plantgenome2016.01.0009.

Avendaño S, Woolliams JA, Villanueva B. 2004. Mendelian sampling terms as a selective advantage in optimum breeding schemes with restrictions on the rate of inbreeding. Genet Res. 83(1):55–64. doi:10.1017/S0016672303006566.

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

Butler DG, Cullis BR, Gilmour AR, Gogel BJ, Thompson R. 2023. ASReml-R Reference Manual Version 4.2. Hempstead: VSN International.

Endelman JB. 2023. Fully efficient, two-stage analysis of multi-environment trials with directional dominance and multi-trait genomic selection. Theoretical and Applied Genetics. 136(4):65. doi:10.1007/s00122-023-04298-x.

Endelman JB, Jannink JL. 2012. Shrinkage estimation of the realized relationship matrix. G3: Genes, Genomes, Genetics. 2(11):1405–1413. doi:10.1534/g3.112.004259.

Endelman JB, Kante M, Lindqvist-Kreuze H, Kilian A, Shannon LM, Caraza-Harter M V., Vaillancourt B, Mailloux K, Hamilton JP, Buell CR. 2024. Targeted genotyping-by-sequencing of potato and data analysis with R/polyBreedR. Plant Genome.:e20484. doi:10.1002/tpg2.20484.

Falconer DS, Mackay TFC. 1996. Introduction to Quantitative Genetics. 4th ed. Essex: Longman Group.

Fu A, Narasimhan B, Boyd S. 2020. CVXR: An R package for disciplined convex optimization. J Stat Softw. 94(14):1–34. doi:10.18637/jss.v094.i14.

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

Gaynor RC, Gorjanc G, Bentley AR, Ober ES, Howell P, Jackson R, Mackay IJ, Hickey JM. 2017. A two-part strategy for using genomic selection to develop inbred lines. Crop Sci. 57(5):2372–2386. doi:10.2135/cropsci2016.09.0742.

Gaynor RC, Gorjanc G, Hickey JM. 2021. AlphaSimR: An R package for breeding program simulations. G3: Genes, Genomes, Genetics. 11(2):jkaa017. doi:10.1093/G3JOURNAL/JKAA017.

Gorjanc G, Hickey JM. 2018. AlphaMate: A program for optimizing selection, maintenance of diversity and mate allocation in breeding programs. Bioinformatics. 34(19):3408–3411. doi:10.1093/bioinformatics/bty375.

Grundy B, Villanueva B, Woolliams JA. 1998. Dynamic selection procedures for constrained inbreeding and their consequences for pedigree development. Genet Res. 72(2):159–168. doi:10.1017/S0016672398003474.

Henryon M, Liu H, Berg P, Su G, Nielsen HM, Gebregiwergis GT, Sørensen AC. 2019. Pedigree relationships to control inbreeding in optimum-contribution selection realise more genetic gain than genomic relationships. Genetics Selection Evolution. 51(1):39. doi:10.1186/s12711-019-0475-5.

Holland JH. 1992. Genetic algorithms. Scientific American. 267:66–73.

Ibe OC. 2014. Fundamentals of Applied Probability and Random Processes. San Diego: Academic Press.
Kante M, Lindqvist-Kreuze H, Portal L, David M, Gastelo M. 2021. Kompetitive Allele Specific PCR (KASP) markers for potato: An effective tool for increased genetic gains. Agronomy. 11:2315. doi:10.3390/agronomy11112315.
Kerr RJ, Li L, Tier B, Dutkowski GW, McRae TA. 2012. Use of the numerator relationship matrix in genetic analysis of autopolyploid species. Theoretical and Applied Genetics. 124(7):1271–1282. doi:10.1007/s00122-012-1785-y.
Kinghorn BP. 2011. An algorithm for efficient constrained mate selection. Genetics Selection Evolution. 43(1):4. doi:10.1186/1297-9686-43-4.
Labroo MR, Endelman JB, Gemenet DC, Werner CR, Gaynor RC, Covarrubias-Pazaran GE. 2023. Clonal diploid and autopolyploid breeding strategies to harness heterosis: insights from stochastic simulation. Theoretical and Applied Genetics. 136(7):147. doi:10.1007/s00122-023-04377-z.
Labroo MR, Studer AJ, Rutkoski JE. 2021. Heterosis and Hybrid Crop Breeding: A Multidisciplinary Review. Front Genet. 12:643761. doi:10.3389/fgene.2021.643761.
Meuwissen T, Goddard M. 2010. The use of family relationships and linkage disequilibrium to impute phase and missing genotypes in up to whole-genome sequence density genotypic data. Genetics. 185(4):1441–1449. doi:10.1534/genetics.110.113936.
Meuwissen THE. 1997. Maximizing the Response of Selection with a Predefined Rate of Inbreeding. J Anim Sci. 75:934–940.
Meuwissen THE, Sonesson AK, Gebregiwergis G, Woolliams JA. 2020. Management of Genetic Diversity in the Era of Genomics. Front Genet. 11:880. doi:10.3389/fgene.2020.00880.
MOSEK ApS. 2024. The R to MOSEK Optimization Interface. https://cran.r-project.org/web/packages/Rmosek
Pesek J, Baker RJ. 1969. Desired improvement in relation to selection indices. Canadian Journal of Plant Science. 49:803–804.
Pong-Wong R, Woolliams JA. 2007. Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming. Genetics Selection Evolution. 39(1):3–25. doi:10.1051/gse:2006031.
Powell JE, Visscher PM, Goddard ME. 2010. Reconciling the analysis of IBD and IBS in complex trait studies. Nat Rev Genet. 11(11):800–805. doi:10.1038/nrg2865.
Sonesson AK, Woolliams JA, Meuwissen THE. 2012. Genomic selection requires genomic control of inbreeding. Genetics Selection Evolution. 44(1):27. doi:10.1186/1297-9686-44-27.
Toro M, Pérez-Enciso M. 1990. Optimization of selection response under restricted inbreeding. Genetics Selection Evolution. 22:93–107.
Toro MA, Varona L. 2010. A note on mate allocation for dominance handling in genomic selection. Genetics Selection Evolution. 42:33.
VanRaden PM. 2008. Efficient methods to compute genomic predictions. J Dairy Sci. 91(11):4414–23. doi:10.3168/jds.2007-0980.
Vitezica ZG, Varona L, Legarra A. 2013. On the additive and dominant variance and covariance of individuals within the genomic selection scope. Genetics. 195(4):1223–1230. doi:10.1534/genetics.113.155176.
Wellmann R. 2019. Optimum contribution selection for animal breeding and conservation: The R package optiSel. BMC Bioinformatics. 20(1):25. doi:10.1186/s12859-018-2450-5.

Werner CR, Gaynor RC, Sargent DJ, Lillo A, Gorjanc G, Hickey JM. 2023. Genomic selection strategies for clonally propagated crops. Theoretical and Applied Genetics. 136(4):74. doi:10.1007/s00122-023-04300-6.

Whalen A, Ros-Freixedes R, Wilson DL, Gorjanc G, Hickey JM. 2018. Hybrid peeling for fast and accurate calling, phasing, and imputation with sequence data of any coverage in pedigrees. Genetics Selection Evolution. 50(1). doi:10.1186/s12711-018-0438-2.

Woolliams JA, Berg P, Dagnachew BS, Meuwissen THE. 2015. Genetic contributions and their optimization. Journal of Animal Breeding and Genetics. 132(2):89–99. doi:10.1111/jbg.12148.

Wray NR, Goddard ME. 1994. Increasing long-term response to selection. Genetics Selection Evolution. 26:431–451.

Wray NR, Thompson R. 1990. Prediction of rates of inbreeding in selected populations. Genet Res. 55(1):41–54. doi:10.1017/S0016672300025180.

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

Zheng C, Amadeu RR, Munoz PR, Endelman JB. 2021. Haplotype reconstruction in connected tetraploid F1 populations. Genetics. 219(2). doi:10.1093/genetics/iyab106.

TABLES

Table 1. Parameters of the quadratic equation for dominance values, $Ap_1^2 + Bp_1p_2 + Cp_2^2 + Dp_1 + Ep_2 + F$.

	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. ANOVA p-values for the effect of kinship (A vs. G_{IBD}) and inbreeding constraint (EQ vs. UB) on the components of genetic variance after one generation of COMA selection.

Effect	Df	Additive Genic Variance	Additive Disequilibrium Covariance	Dominance Genic Variance	Dominance Disequilibrium Covariance
Replicate	9	$< 2 \times 10^{-16}$	0.03	$< 2 \times 10^{-16}$	0.02
Constraint	1	0.45	0.85	0.79	4×10^{-8}
Kinship	1	0.0001	0.09	0.55	0.37
Constraint:Kinship	1	0.82	0.14	0.96	0.01
Residuals	27				

Table 3. The influence of constraints for potato virus Y resistance (PVYR) on the COMA solution with $\Delta F_1 = 0.5\%$ and $\Delta F_2 = 1.0\%$.

PVYR Constraint	Response (σ_A)	# Parents	# Matings	PVYR freq
None	1.23	43	43	8.6%
At least one R parent per mating	1.02	42	40	12.5%
PVYR freq $\geq 12.5\%$	1.07	43	42	12.5%

FIGURES

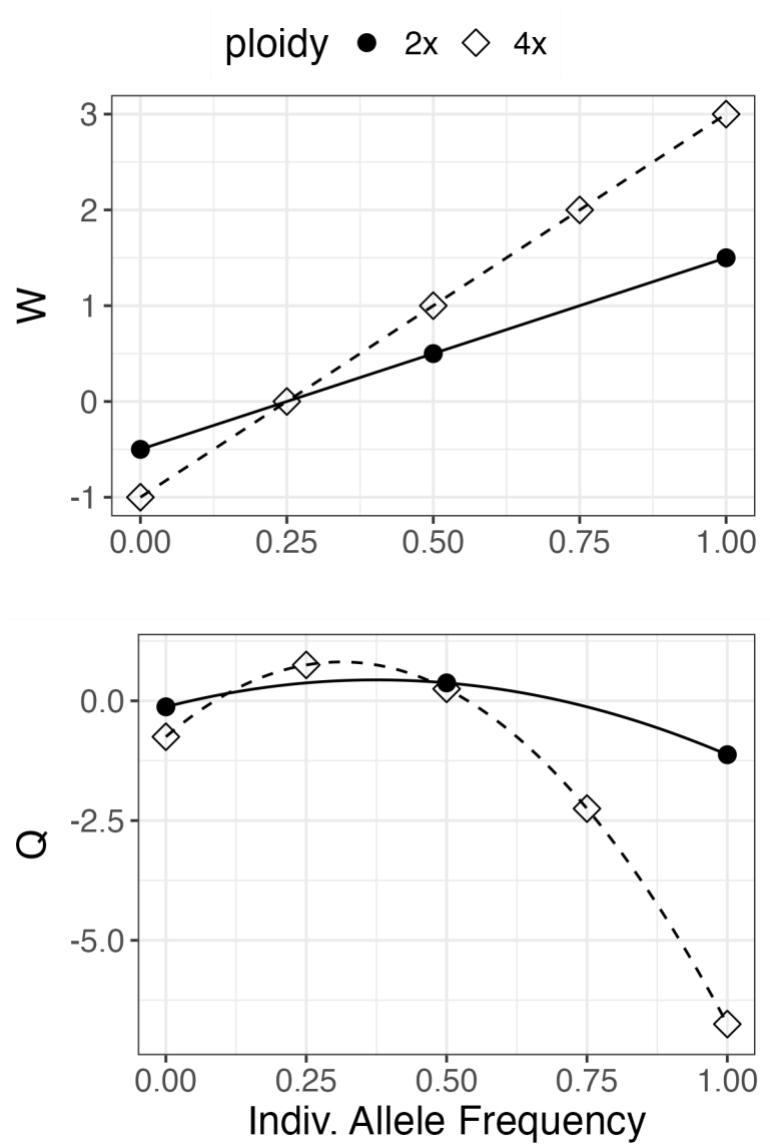


Figure 1. The additive (W) and dominance (Q) regressors as linear vs. quadratic functions, respectively, of individual allele frequency (dosage/ploidy). The population frequency is ¼.

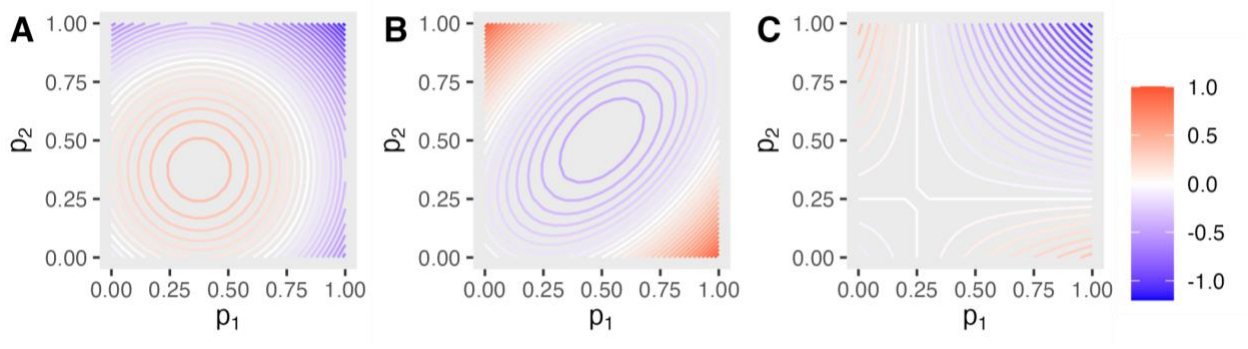


Figure 2. Level curves for mid-parent dominance value (A), mid-parent heterosis (B), and their sum (C), in diploids. The population frequency is ¼.

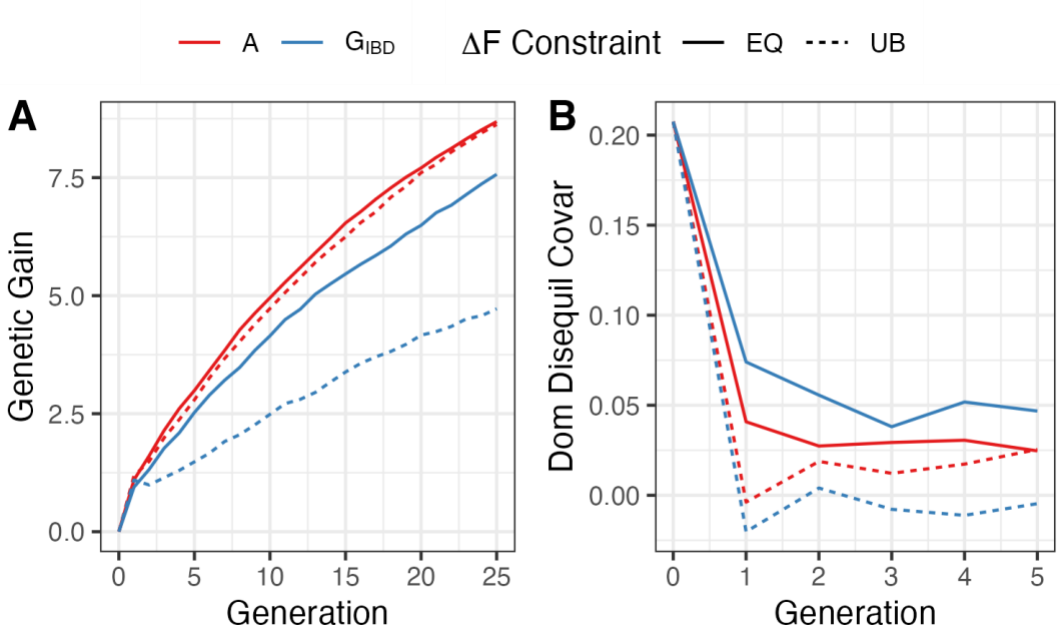


Figure 3. Diploid simulation results for genetic gain and dominance disequilibrium covariance, using COMA with either equality (EQ) or upper bound (UB) constraints for the target inbreeding rate of 0.5%.

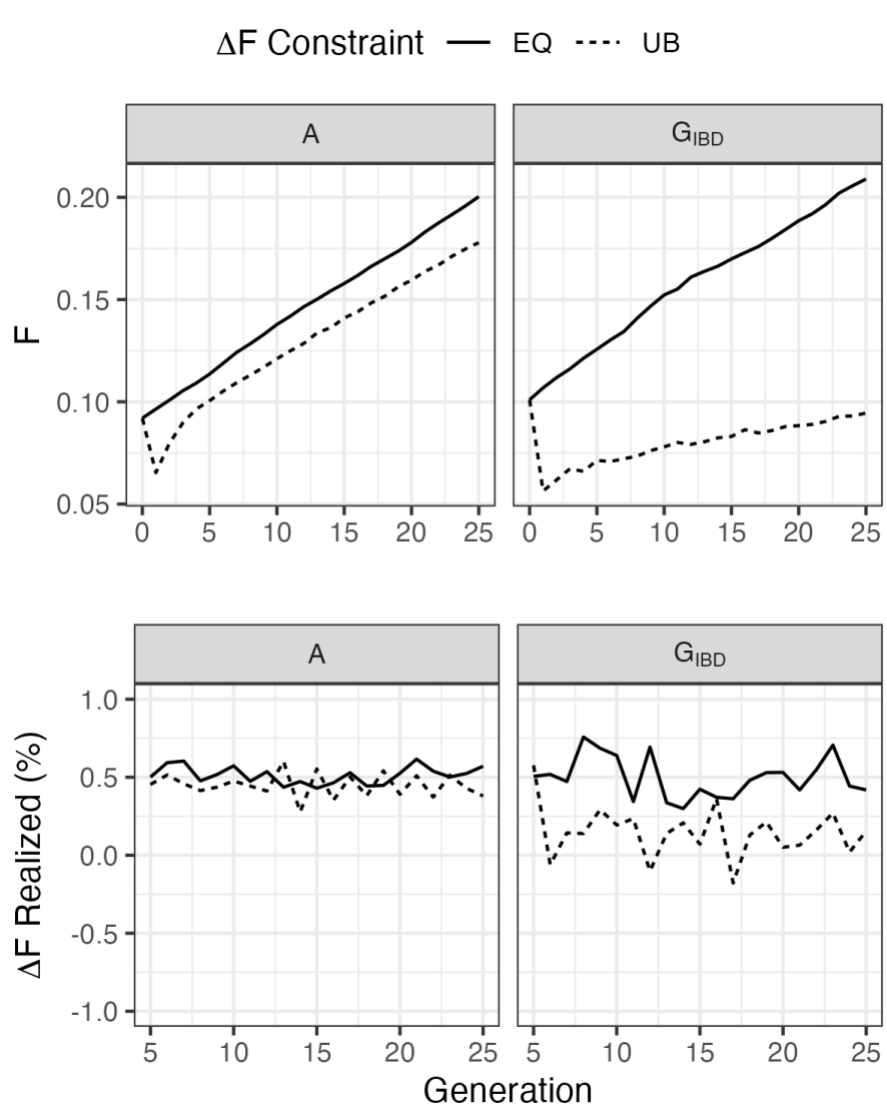


Figure 4. Realized inbreeding coefficients (top) and rates (bottom), using COMA with either equality (EQ) or upper bound (UB) constraints. The target inbreeding rate was 0.5%, based either on pedigree (A) or genomic (G_{IBD}) kinship.

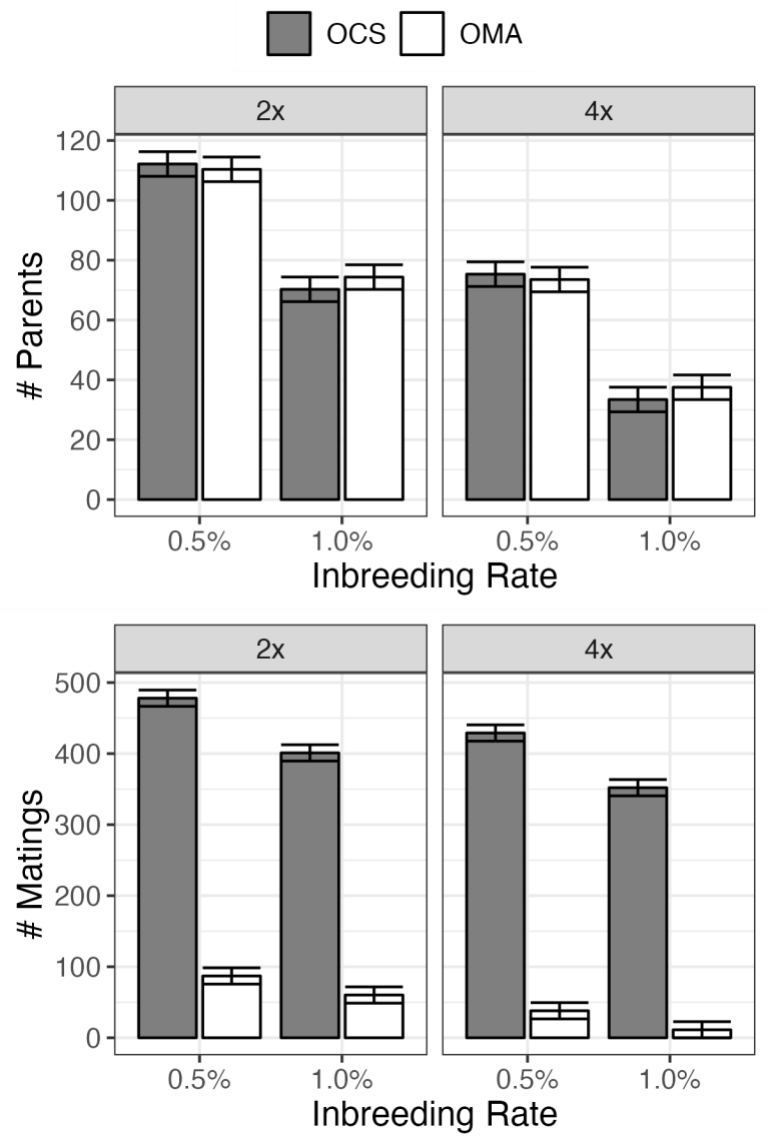


Figure 5. Number of parents and matings for generations 5–25, using either OCS or OMA and pedigree kinship at 0.5% or 1.0% inbreeding. The error bars are the 95% CI based on 10 independent simulations.

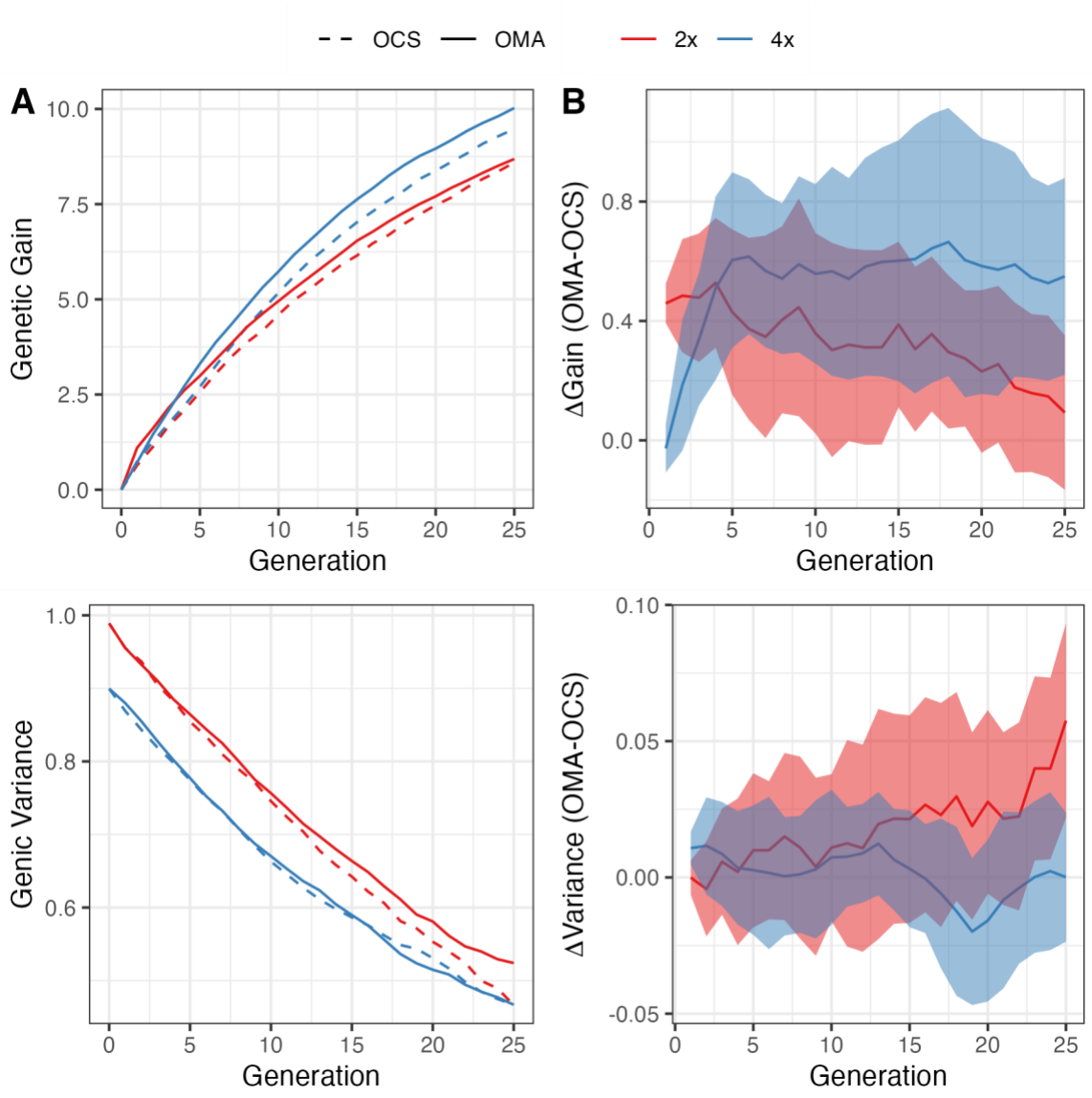


Figure 6. (A) Simulation results for genetic gain and genic variance using OCS vs. OMA at 0.5% inbreeding with pedigree kinship. (B) The solid line is the mean difference across the 10 replicate simulations, and the ribbon is the 95% CI from a paired *t*-test each generation.

.

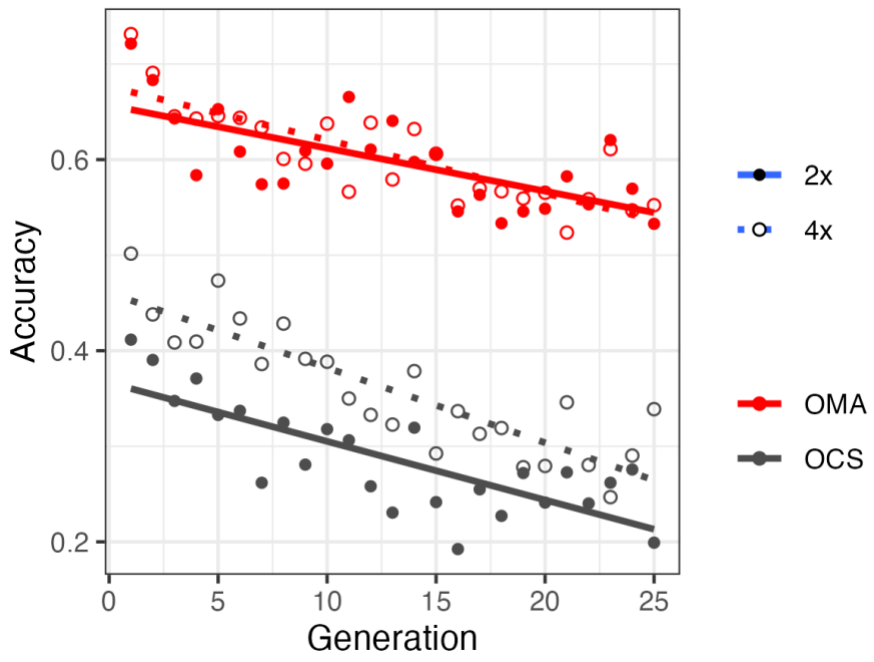


Figure 7. Accuracy of OMA and OCS for predicting mate performance, based on 100 random matings and 50 progeny per mating. Each point is the average of 10 replicate simulations.

File S1: Supplemental Methods

Jeff Endelman

2024-10-03

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. 8), 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 total contribution y_i is the average of the sex-specific contributions (Eq. 8). Eq. 9 follows because breeding value is two times GCA at HWE.

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,

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

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

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

Using the following intermediate results based on Eq. 8,

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

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

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

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 $\mathbf{G}$ (VanRaden Method 1) depend on the individual allele dosages z_{ik}, which have population mean ϕp_k.

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

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

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

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

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

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

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

Substituting the above result into following expression (Eq. 21) demonstrates equality with $\overline{f}_A$.

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

File S2: Supplemental Tables and Figures

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

Table S1. Dominance variance (V_D) and inbreeding depression (b) relative to additive variance (V_A) and standard deviation, respectively.

Dataset	Ploidy	V_D/V_A	$b/\sqrt{V_A}$
Potato	4x	0.38	7.1
Simulation Founders	4x	0.12	7.3
Simulation Founders	2x	0.25	4.3

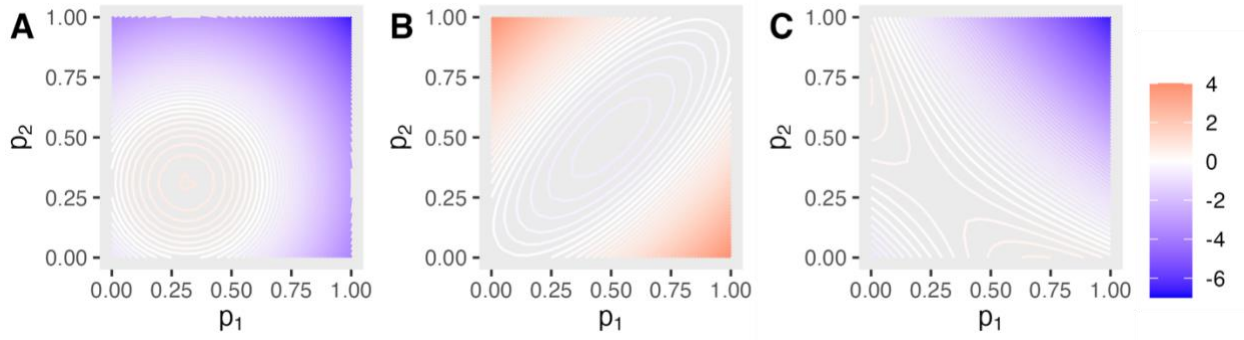


Figure S1. Level curves for mid-parent dominance value (A), mid-parent heterosis (B), and their sum (C), in tetraploids. The population frequency is ¼.

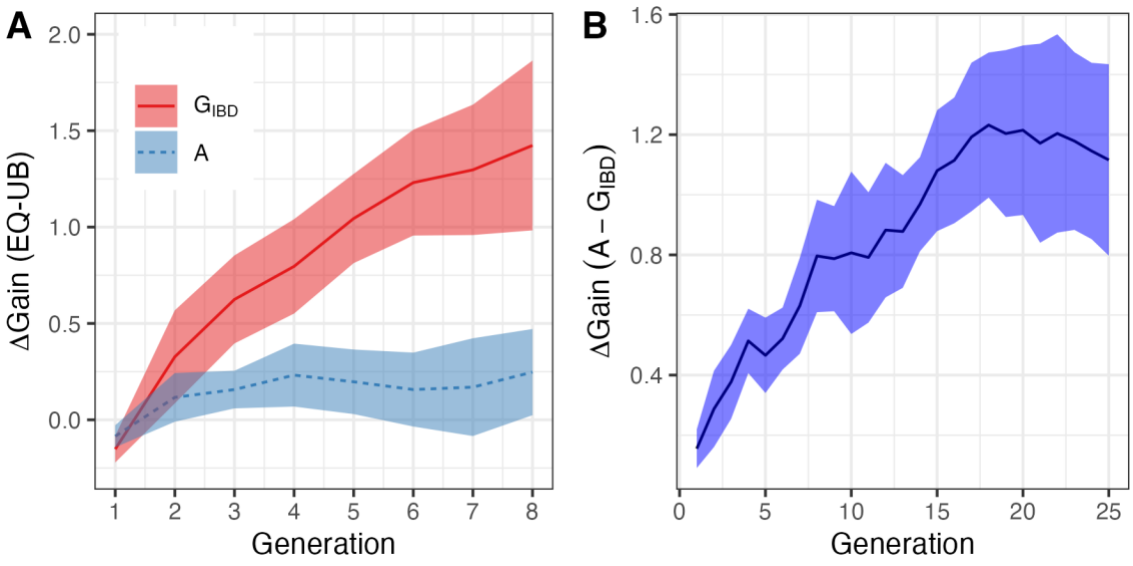


Figure S2. (A) Genetic gain difference at 0.5% inbreeding for equality (EQ) vs. upper bound (UB) constraints with COMA in diploids, using either pedigree or genomic kinship. The solid line is the mean difference across the 10 replicate simulations, and the ribbon is the 95% CI from a paired *t*-test at each generation. (B) Genetic gain difference between pedigree and genomic kinship at 0.5% inbreeding with EQ constraints in diploids.

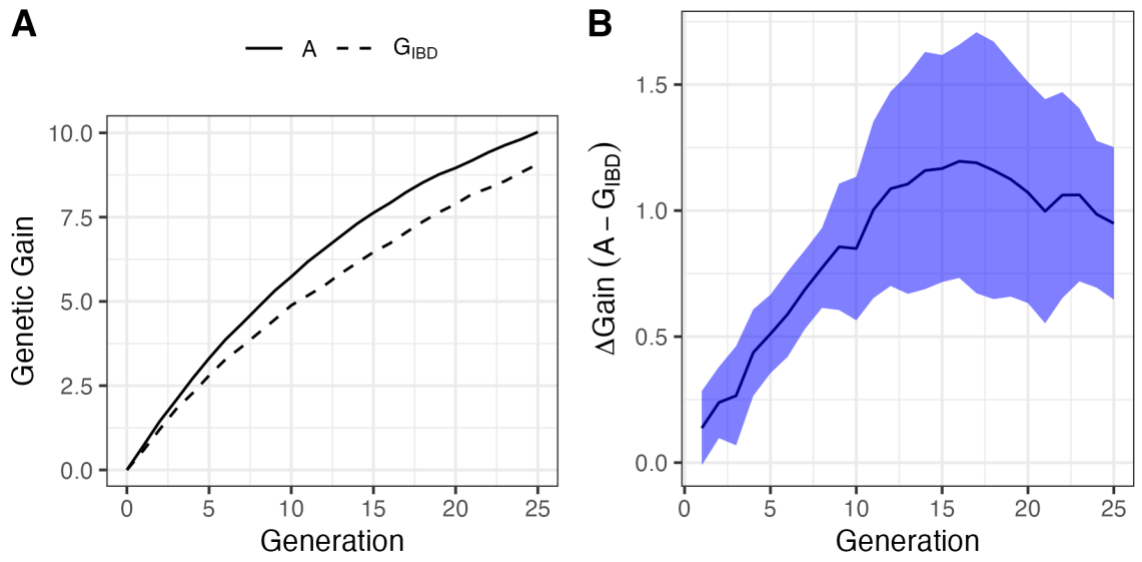


Figure S3. (A) Genetic gain for the 4x simulation at 0.5% inbreeding, using COMA with EQ constraints. (B) Mean difference between the pedigree and genomic kinship results, with the ribbon showing the 95% CI.

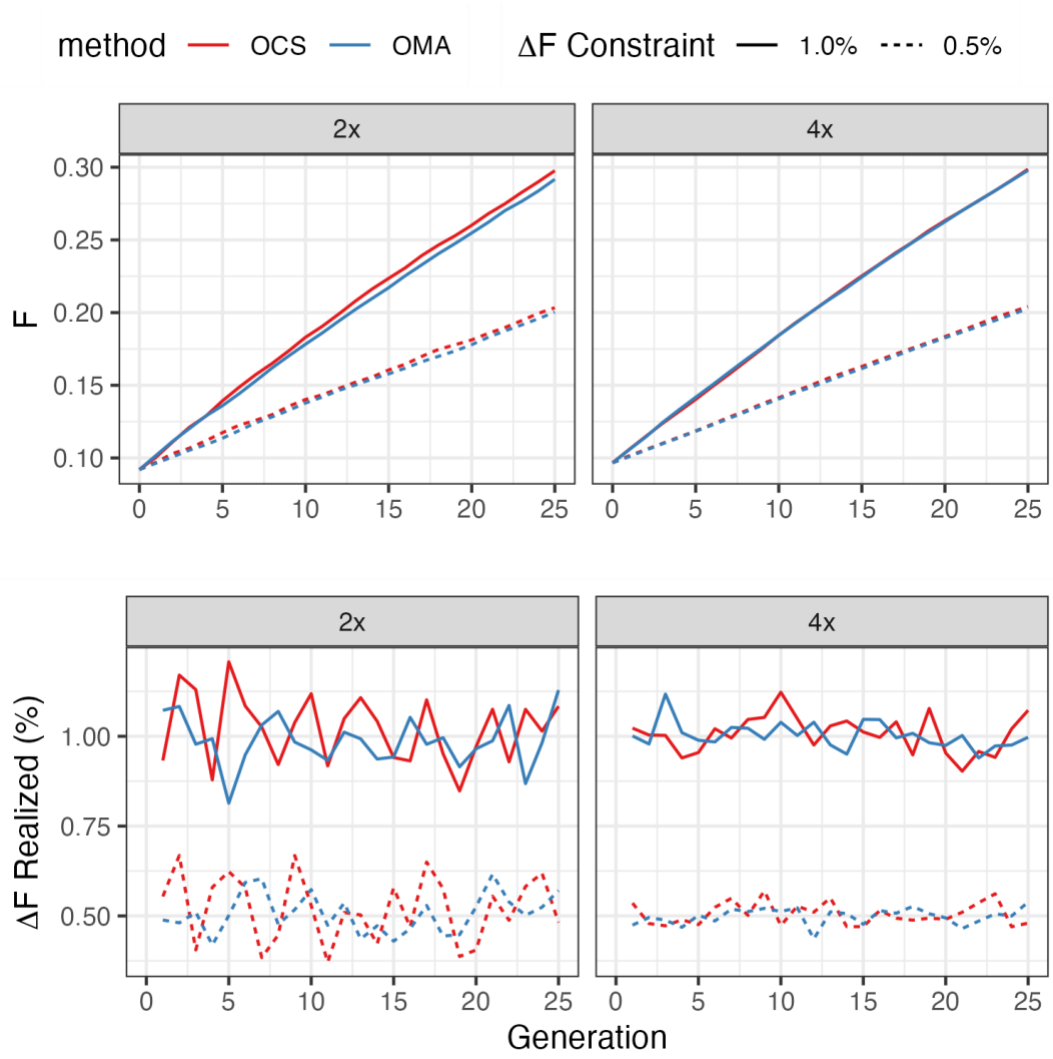


Figure S4. Realized inbreeding coefficients (top) and rates (bottom) at target inbreeding rates of 0.5% and 1.0%. On the left are diploid (2x) results; on the right are tetraploid (4x) results.

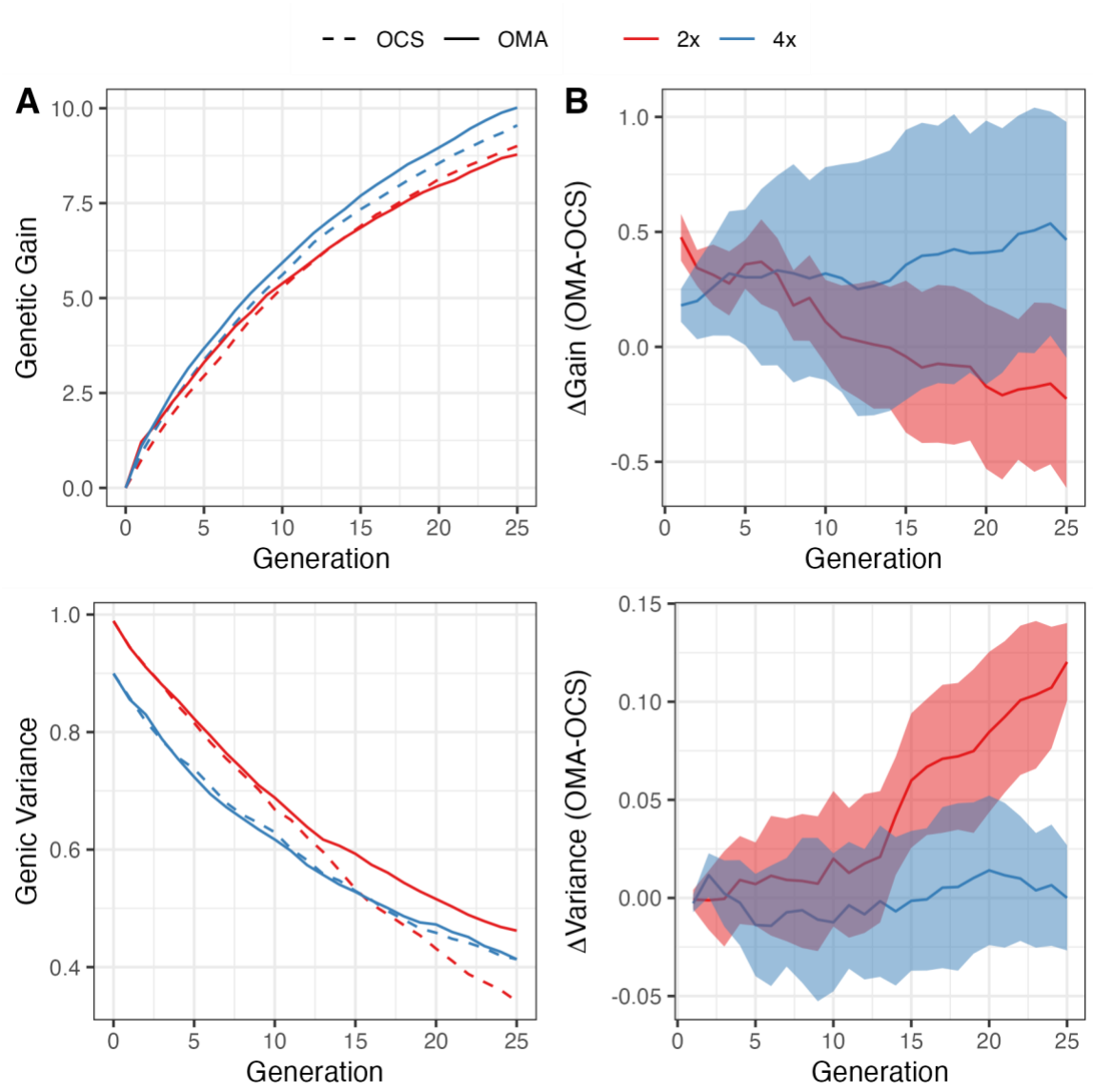


Figure S5. (A) Simulation results for genetic gain and genic variance using OCS vs. OMA at 1.0% inbreeding with pedigree kinship. (B) The solid line is the mean difference across the 10 replicate simulations, and the ribbon is the 95% CI from a paired *t*-test each generation.

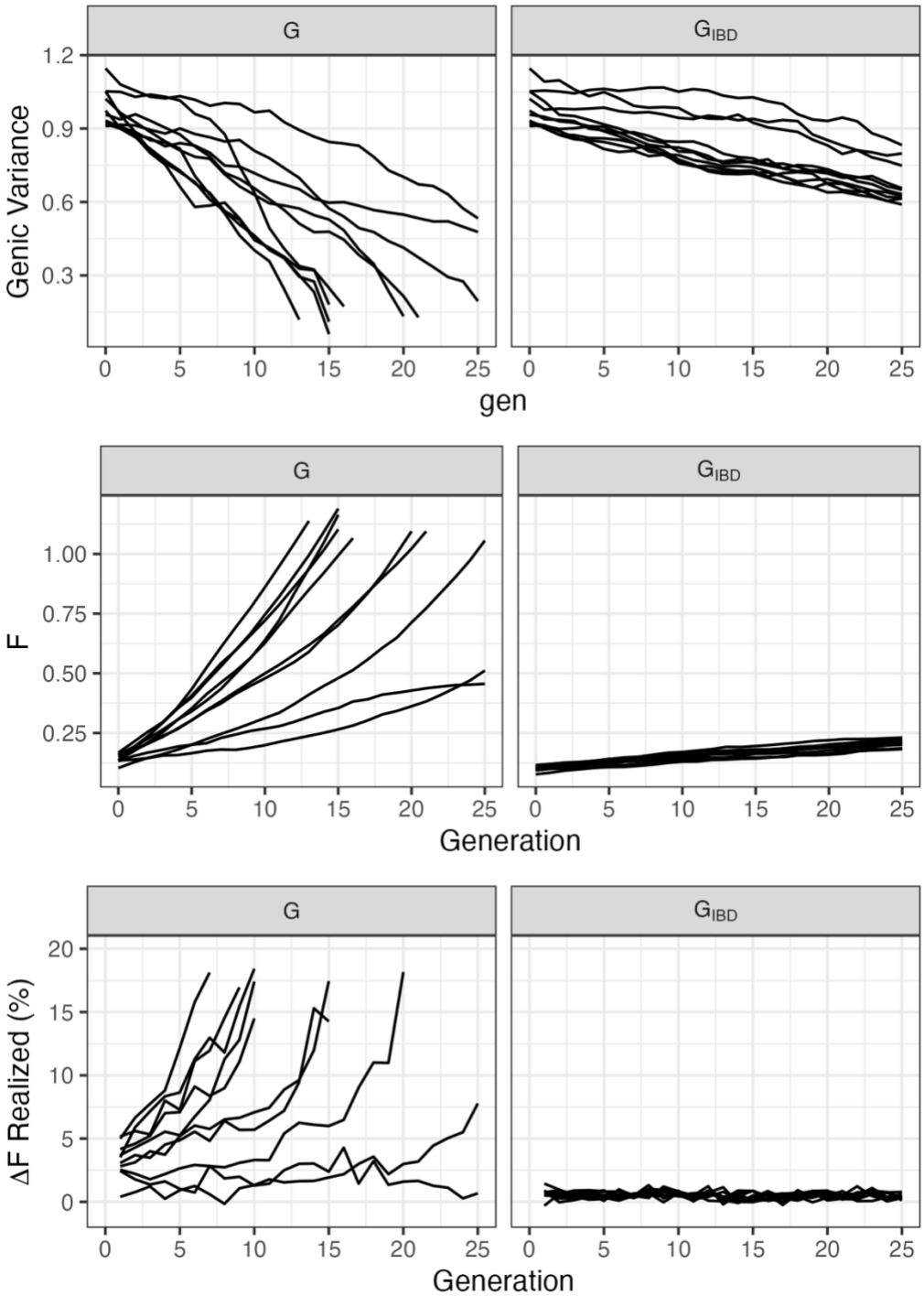


Figure S6. Comparison of **G** (VanRaden Method 1) vs. $\mathbf{G}_{\mathrm{IBD}}$ for genomic control of inbreeding with COMA at target rate 0.5%. Each line is an independent simulation. Inbreeding coefficients for **G** are defined in Eq. 22.

File S3: COMA Vignette

Jeff Endelman

The COMA package implements a new algorithm for optimum mate allocation (OMA), as well as the more traditional method, optimum contribution selection (OCS).

COMA requires two input files. The first 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 dataset comes 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
```

The second input file is the kinship matrix to control inbreeding. Based on simulation results (Endelman 2024), the pedigree IBD kinship matrix is recommended for either OCS or OMA. The kinship matrix for the potato clones was generated 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
```

Now we are ready for `read_data`:

```r
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 -0.6518888
## 2 W17037-10  0.6155129
## 3 W17037-11  0.4235588
## 4 W17037-12  0.8016658
## 5 W17037-13  0.3160459
## 6 W17037-15  0.6178174
```

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

```
##     parent1  parent2       merit
## 1  W17037-1 W17037-1 -2.06643337
## 2 W17037-10 W17037-1 -0.47212478
## 3 W17037-11 W17037-1 -0.42772748
## 4 W17037-12 W17037-1 -0.08777775
## 5 W17037-13 W17037-1 -0.24849597
## 6 W17037-15 W17037-1 -0.32733990
```

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 `standardize`=TRUE, genetic values are scaled by the additive standard deviation (σ_A). This is for convenience when interpreting the response values.

As shown above, the `parents` output contains the predicted merit (GEBV) for each individual. The `matings` output contains the predicted merit (F1 progeny mean) for each possible mating. 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".

OMA and OCS

The first argument of the `oma` function is `dF`=c(min,max), which represents lower and upper bounds on the inbreeding rate for the next generation, dF1. The upper bound is also applied to the average inbreeding rate over two generations, dF2, assuming random mating of the progeny. The second argument, `parents`=data.frame(id,min,max), contains lower and upper bounds on the contribution of each parent (do not include a column for parental merit). The third argument, `matings`, is a data frame of the merit for each possible mating, as well as lower and upper bounds on the allocation.

```r
parents <- data.frame(id=data$parents$id, min=0, max=1)
matings <- data.frame(data$matings, min=0, max=1)
ans0 <- oma(dF=c(-1,0.005), parents=parents, matings=matings, ploidy=4, K=data$K)
ans0$response
```

```
##   dF1 dF2 merit n.parent n.mate
## 1  NA  NA    NA        0      0
```

The above result shows there is no feasible solution at 0.5% inbreeding for both generations, due to the relatedness of the clones. The argument `dF.adapt` can be used to automatically increasing the upper bound until a solution is found:

```r
ans1 <- oma(dF=c(-1,0.005), parents=parents, matings=matings, ploidy=4, K=data$K,
            dF.adapt=list(step=0.001,max=0.05))
ans1$response
```

```
##       dF1   dF2    merit n.parent n.mate
## 1 -0.0082 0.008 1.215909       33     31
```

With no lower bound, a solution was found at dF2 = 0.8%, while the negative value for dF1 indicates the progeny would be 0.8% more outbred than the parents. When a similar approach was used in a simulation study (Endelman 2024), the population became trapped in a local optimum with limited long-term gain. Imposing a lower bound of 0.5% for dF1 led to more long-term gain in the simulation. As the following code shows, for the potato dataset, this required increasing the dF2 rate to 1.0%.

```r
ans2 <- oma(dF=c(0.005,0.008), parents=parents, matings=matings, ploidy=4, K=data$K,
            dF.adapt=list(step=0.001,max=0.05))
ans2$response
```

```
##     dF1  dF2    merit n.parent n.mate
## 1 0.005 0.01 1.228329       43     43
```

Interestingly, the second solution uses more parents even though the inbreeding rate is higher. The optimal parental contributions are returned as `oc`, and the optimal mate allocations are returned as `om`.

```r
head(ans2$oc)
```

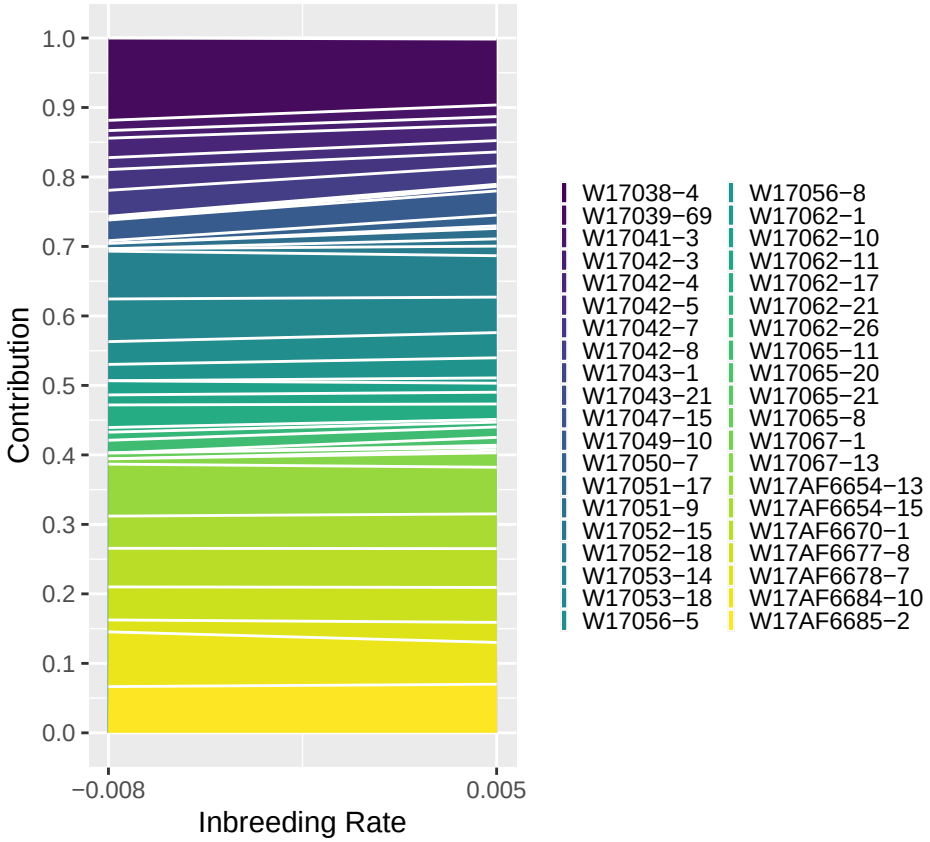
```
##           id        value
## 40  W17038-4 0.0015535826
## 43 W17039-11 0.0008559821
## 71 W17039-69 0.0944907070
## 77  W17041-3 0.0167267011
## 78  W17042-3 0.0116170003
## 79  W17042-4 0.0227763475
```

```r
head(ans2$om)
```

```
##           parent1   parent2        value
## 5987    W17062-26  W17038-4 0.0031071652
## 6374    W17062-26 W17039-11 0.0017119643
## 9497     W17042-8 W17039-69 0.0292801673
## 9581  W17AF6677-8 W17039-69 0.0197916263
## 9583 W17AF6684-10 W17039-69 0.0004465966
## 9584  W17AF6685-2 W17039-69 0.1394630237
```

The parental contributions of the two solutions can be visually compared with the COMA function `plot_ribbon`:

```r
library(ggplot2)
plot_ribbon(oc=list(`-0.008`=ans1$oc,`0.005`=ans2$oc)) +
   theme(legend.key.size=unit(0.1,'cm'))
```



COMA also has a function for OCS, which optimizes parental contributions but not specific matings. The arguments are similar, except dF is only a single number for the upper bound on group kinship, and the `parents` data frame includes a column for merit:

```
par2 <- data.frame(data$parents, min=0, max=1)
ans3 <- ocs(dF=0.005, parents=par2, ploidy=4, K=data$K,
            dF.adapt=list(step=0.001, max=0.05))
ans3$response
```

```
##      dF    merit n.parent
## 1 0.015 1.222502       29
```

The smallest feasible inbreeding rate under OCS was 1.5%. The predicted merit is based on parental breeding values and therefore neglects specific combining ability (SCA). When SCA was accounted for, the OCS solution (using random mating) decreased from 1.2 to 1.0 σ_A.

```
ocs.matings <- merge(expand.grid(parent1=ans3$oc$id,parent2=ans3$oc$id,stringsAsFactors = F),
                     data$matings)
i1 <- match(ocs.matings$parent1,ans3$oc$id)
i2 <- match(ocs.matings$parent2,ans3$oc$id)
ocs.matings$value <- ifelse(i1==i2,ans3$oc$value[i1]*ans3$oc$value[i1],
                            2*ans3$oc$value[i1]*ans3$oc$value[i2])
sum(ocs.matings$merit*ocs.matings$value)
```

```
## [1] 1.038667
```

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 with the 0.5% lower bound:

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

#calculate proportion of matings with PVYR parent from previous solution
n.mate <- ans2$response$n.mate
nR <- sum(ans2$om$parent1 %in% PVYR.id | ans2$om$parent2 %in% PVYR.id)

#add constraint
ans2a <- oma(dF=c(0.005,0.01), ploidy=4L, K=data$K, parents=parents,
            matings=filter(matings, parent1%in%PVYR.id | parent2%in%PVYR.id))

#calculate frequency of PVYR gene, assuming single copy in parents
R.parents <- which(ans2$oc$id %in% PVYR.id)
freq2 <- sum(ans2$oc$value[R.parents])*0.25

R.parents <- which(ans2a$oc$id %in% PVYR.id)
freq2a <- sum(ans2a$oc$value[R.parents])*0.25

result <- data.frame(constraint=c("none","method 1"),
                     rbind(ans2$response,ans2a$response),
                     prop.resistant.matings=c(nR/n.mate,1),
                     Ry.freq=c(freq2,freq2a))
kable(result,digits=3)
```

constraint	dF1	dF2	merit	n.parent	n.mate	prop.resistant.matings	Ry.freq
none	0.005	0.01	1.228	43	43	0.442	0.086
method 1	0.005	0.01	1.018	42	40	1.000	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, $\mathbf{s}$, 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:

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

```r
par3 <- data.frame(parents,
                   gt0.125=ifelse(parents$id %in% PVYR.id, 0.25, 0))
ans2b <- oma(dF=c(0.005,0.01), ploidy=4L, K=data$K, parents=par3, matings=matings)

nR <- sum(ans2b$om$parent1 %in% PVYR.id | ans2b$om$parent2 %in% PVYR.id)
R.parents <- which(ans2b$oc$id %in% PVYR.id)
freq2b <- sum(ans2b$oc$value[R.parents])*0.25

result <- rbind(result,
                data.frame(constraint="method 2",ans2b$response,
                           prop.resistant.matings=nR/n.mate, Ry.freq=freq2b))

kable(result,digits=3)
```

constraint	dF1	dF2	merit	n.parent	n.mate	prop.resistant.matings	Ry.freq
none	0.005	0.01	1.228	43	43	0.442	0.086
method 1	0.005	0.01	1.018	42	40	1.000	0.125
method 2	0.005	0.01	1.070	43	42	0.628	0.125

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

Package ‘COMA’

October 1, 2024

Title Convex Optimization of Mate Allocation

Version 0.19

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 3
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 increase dF by dF.adapt$step up to dF.adapt$max.

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

Originally, only an upper bound was placed on the average kinship of the progeny, according to `dF`. As of package v0.19, both lower and upper bounds are allowed by passing a vector of length 2. When a single number is supplied, it is interpreted as equal lower and upper bounds.

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 increase the upper bound by dF.adapt$step up to dF.adapt$max.

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,
  partition = FALSE
)
```

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 based on additive std. dev.
`n.core`	multi-core evaluation
`partition`	T/F, partition matings as MPA, MPD, MPH for dominance model

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

Index

ocs, 1
oma, 2

plot_ribbon, 3

read_data, 4

File S4: SimPlus Vignette

Jeff Endelman

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)
}

results.all <- 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.all, 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.all, data.frame(results[5:20,], method="OCS-G.IBD-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`. The dF argument for `sim_OMA` is a vector of 2 numbers for the lower and upper bounds. In this case, the numbers are identical to create an equality constraint.

```
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=c(0.01,0.01), 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.all, data.frame(results[5:20,], method="OMA-A-GPMP"))
```

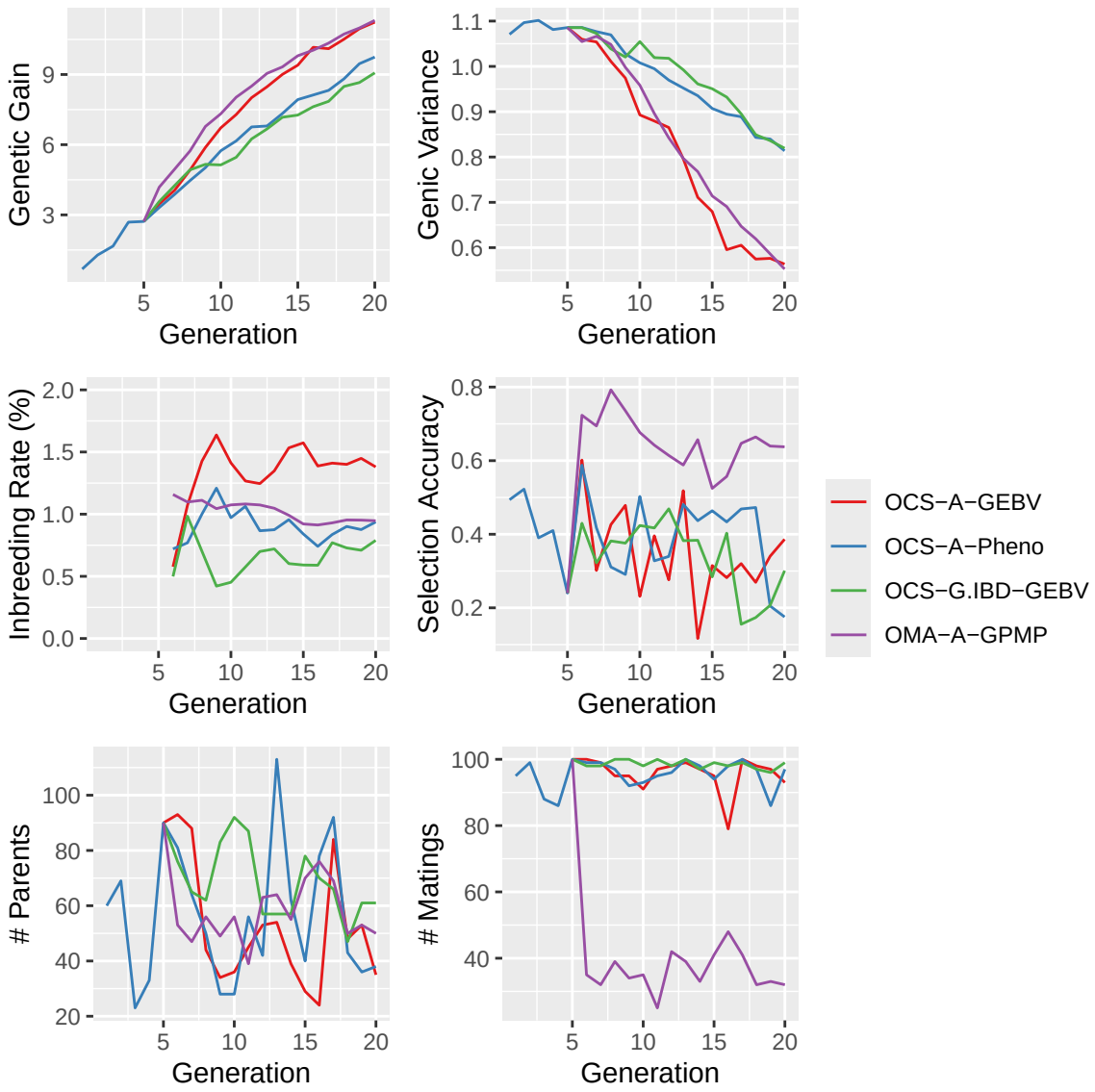
The average inbreeding rate at time t follows

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

```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$dF[ix] <- 100*fdF(gen=1:15,
                              Ft=results.all$F.coeff[ix],
                              F0=results0$F.coeff[results0$method==methods[i]])
}
```

We can now plot the results.

```r
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") + ylim(0,2) +
  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 mate performance and sparser mating designs. OMA generated more gain in the short term, but long-term gains were similar.

Package ‘SimPlus’

October 4, 2024

Title Functions to enhance AlphaSimR

Version 0.06

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

Details

Argument `dF` is a numeric vector of length 2, for lower and upper bounds on inbreeding rate.

Parent pre-selection is done with `ocs` to satisfy the `max.parent` limit for `oma`.

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

list of dominance parameters

`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, G.IBD
`K.file`	kinship CSV filename
`ped.file`	pedigree CSV filename
`p.ref`	reference pop allele frequencies for "G" 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. `K.method="G"` specifies VanRaden Method 1, and 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

Index

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

File S5: Potato Marker Effects

Jeff Endelman

2024-07-27

```
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",workspace="1000mb")
prep1 <- blup_prep(ans1$blues, vcov=ans1$vcov, geno=geno,
                   vars=ans2$vars)
```

Yield and fry color have similar importance as breeding objectives, while later maturity is undesirable. Initially, a restricted index was calculated, with equal merit for yield and fry color and zero gain for maturity.

```
coeff <- c(total.yield=1, vine.maturity=0, fry.color=1)
gain1 <- gain(prep1, merit=coeff, gamma=1,
              restricted=data.frame(trait="vine.maturity", sign="<"))
kable(gain1$table)
```

trait	response	index
total.yield	0.256	0.636
vine.maturity	0.000	-0.436
fry.color	0.436	0.636

Due to the unfavorable correlation between yield and late maturity, the yield response was only 65% of the response for fry color. As an alternative, a desired gains index was calculated, based on equal gains for yield and fry color and zero gain for maturity.

```
gain2 <- gain(prep1, desired=coeff, gamma=1)
kable(gain2$table)
```

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

```r
sum(coeff*gain1$table$response)
```

```
## [1] 0.692
```

```r
sum(coeff*gain2$table$response)
```

```
## [1] 0.68
```

Imposing equal gains reduced total merit by less than 1%, and given the uncertainty of the true merit coefficients, this index was selected.

```r
index.coeff <- gain2$table$index
names(index.coeff) <- gain2$table$trait

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

```
##   ratio  estimate
## 1 Vd/Va 0.3823739
## 2 b/SDa 7.0718983
```

```r
#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))

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")
```

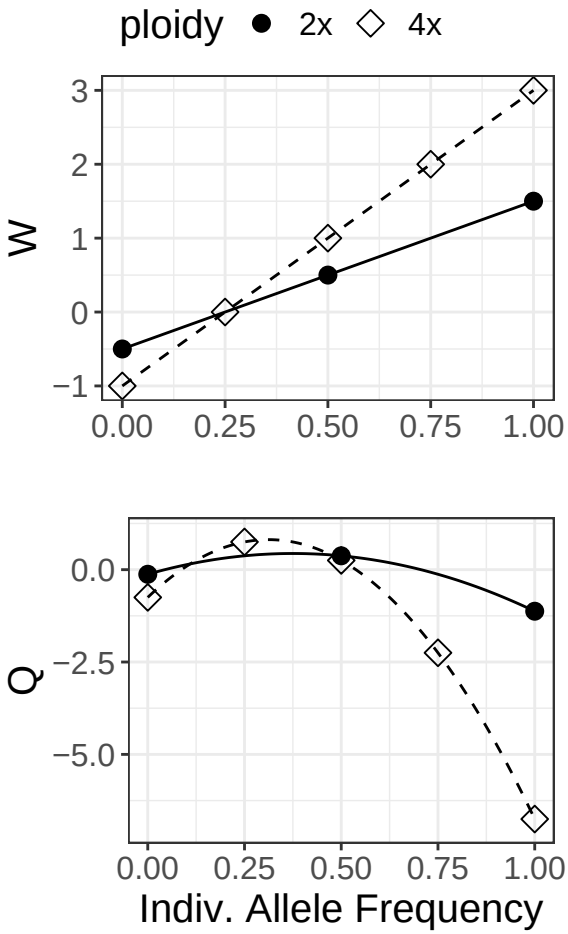
File S6: R Code for Figures

Jeff Endelman

2024-10-06

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("2x",3),rep("4x",5)))

x2 <- seq(0,1,by=0.01)
p1 <- 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="",breaks=c(0:4)/4) + ylab("W") +
  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("2x",3),rep("4x",5)))
p2 <- 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="Indiv. Allele Frequency",breaks=c(0:4)/4) +
  theme(text=element_text(size=15)) + scale_y_continuous(name="Q",limits=c(-7,1))

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

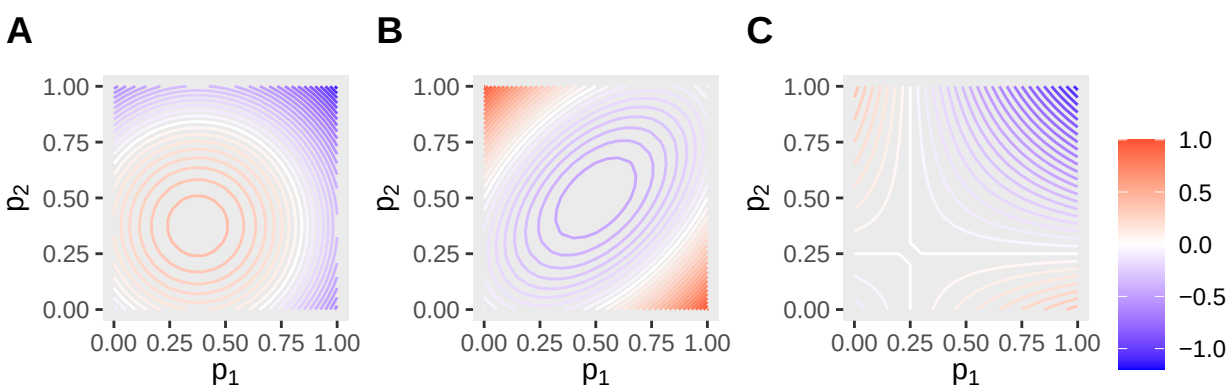



```
ggsave("Figure1.png",path="Figures",device="png",width=4,height=6,units="in")
```

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(p[1])) +
```

```
  ylab(expression(p[2])) + theme(panel.grid=element_blank()) +
  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(p[1])) +
  ylab(expression(p[2])) + theme(panel.grid=element_blank()) +
  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(p[1])) +
  ylab(expression(p[2])) + theme(panel.grid=element_blank()) +
  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)
```

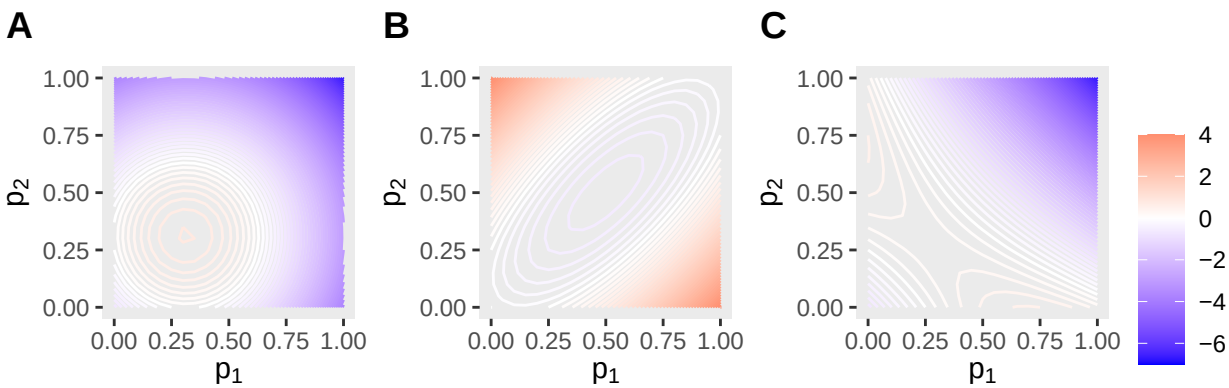



```
ggsave("Figure2.png",path="Figures",device="png",width=7,height=4,units="in")
```

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(p[1])) +
  ylab(expression(p[2])) + theme(panel.grid=element_blank()) +
  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(p[1])) +
  ylab(expression(p[2])) + theme(panel.grid=element_blank()) +
  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(p[1])) +
  ylab(expression(p[2])) + theme(panel.grid=element_blank()) +
  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",path="Figures",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)
```

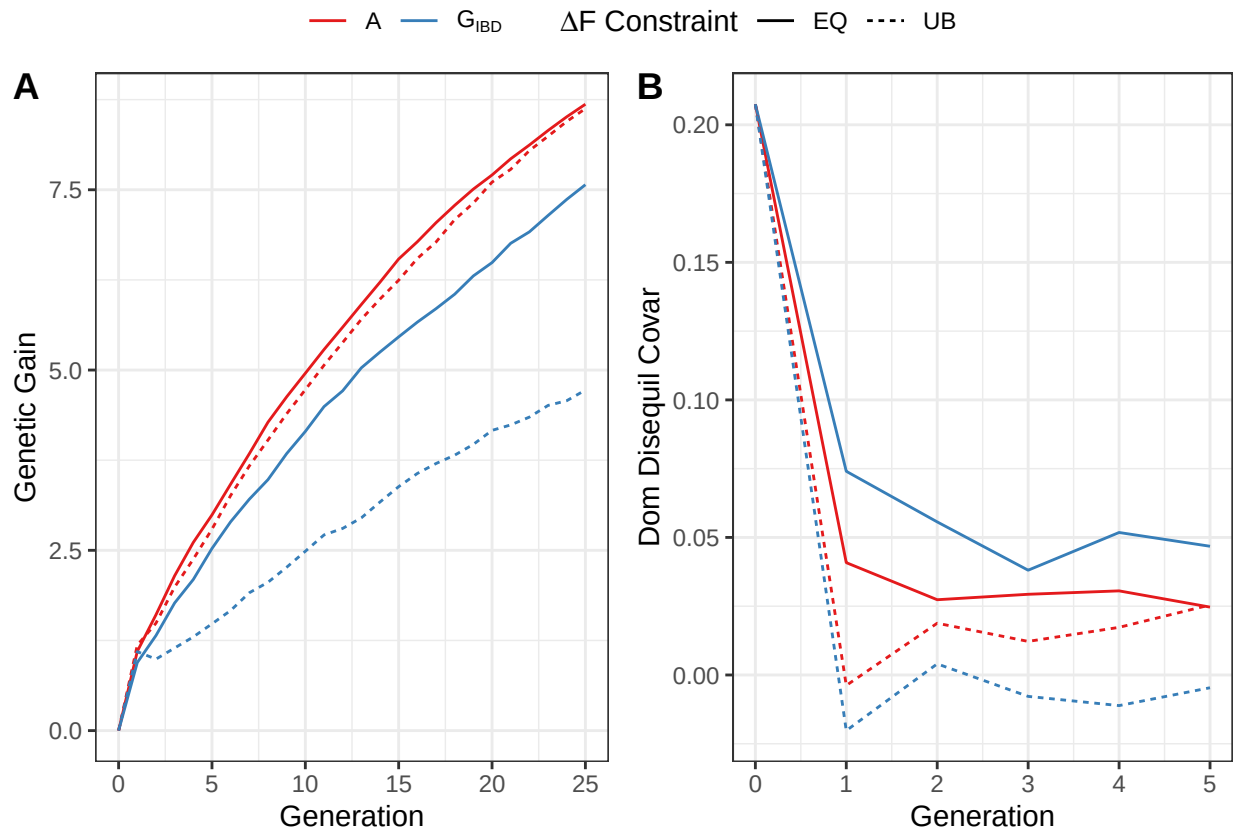
```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(ggplot2)
library(ggpubr)
data <- read.csv("FileS7.csv")
data$genicVar <- data$varA.genic + data$varD.genic
```

```r
data$varD.diseq <- data$varD - data$varD.genic
data$varA.diseq <- data$varA - data$varA.genic
data$varG <- data$varA + data$varD
data$replicate <- factor(data$replicate)
data$dF <- factor(ifelse(data$dF=="0.005","0.5%","1.0%"),ordered=T,
                  levels=c("1.0%","0.5%"))
data$sim <- apply(data[,c("ploidy","kinship","method","dF","replicate")],
                  1,paste,collapse="_")
sims <- unique(data$sim)
nsim <- length(sims)
n <- nrow(data)
data$dFr <- 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])
  n <- length(ix)
  k <- which(data$gen[ix]==0)
  data$gain[ix] <- data$mu[ix] - data$mu[ix[k]]
  data$dFr[ix] <- c(NA,diff(data$F.coeff[ix])/(1-data$F.coeff[ix][-n]))*100
}
```

Figure 3

```r
data2 <- filter(data, ploidy=="2x",dF=="0.5%",method %in% c("OMA","OMA.UB"),
                kinship %in% c("A","G.IBD"))
data2$constraint <- ifelse(data2$method=="OMA.UB","UB","EQ")
data3 <- aggregate(gain~gen+constraint+kinship,data=data2,FUN=mean)

p1 <- ggplot(data3,aes(x=gen,y=gain,linetype=constraint,colour=kinship)) +
  geom_line() + theme_bw() +
  scale_colour_brewer(palette="Set1",name="",labels=c("A",expression(paste(G[IBD])))) +
  ylab("Genetic Gain") + xlab("") + theme(legend.position = "top") +
  scale_linetype(name=expression(paste(Delta,"F Constraint"))) +
  xlab("Generation")

data2 <- filter(data2,gen<=5)
data3 <- aggregate(varD.diseq~gen+constraint+kinship,data=data2,FUN=mean)
p2 <- ggplot(data3,aes(x=gen,y=varD.diseq,linetype=constraint,colour=kinship)) +
  geom_line() + theme_bw() +
  scale_colour_brewer(palette="Set1",name="",labels=c("A",expression(paste(G[IBD])))) +
  ylab("Dom Disequil Covar") + xlab("") + theme(legend.position = "top") +
  scale_linetype(name=expression(paste(Delta,"F Constraint"))) +
  xlab("Generation")

ggarrange(p1,p2,nrow=1,common.legend=T,labels=c("A","B"))
```



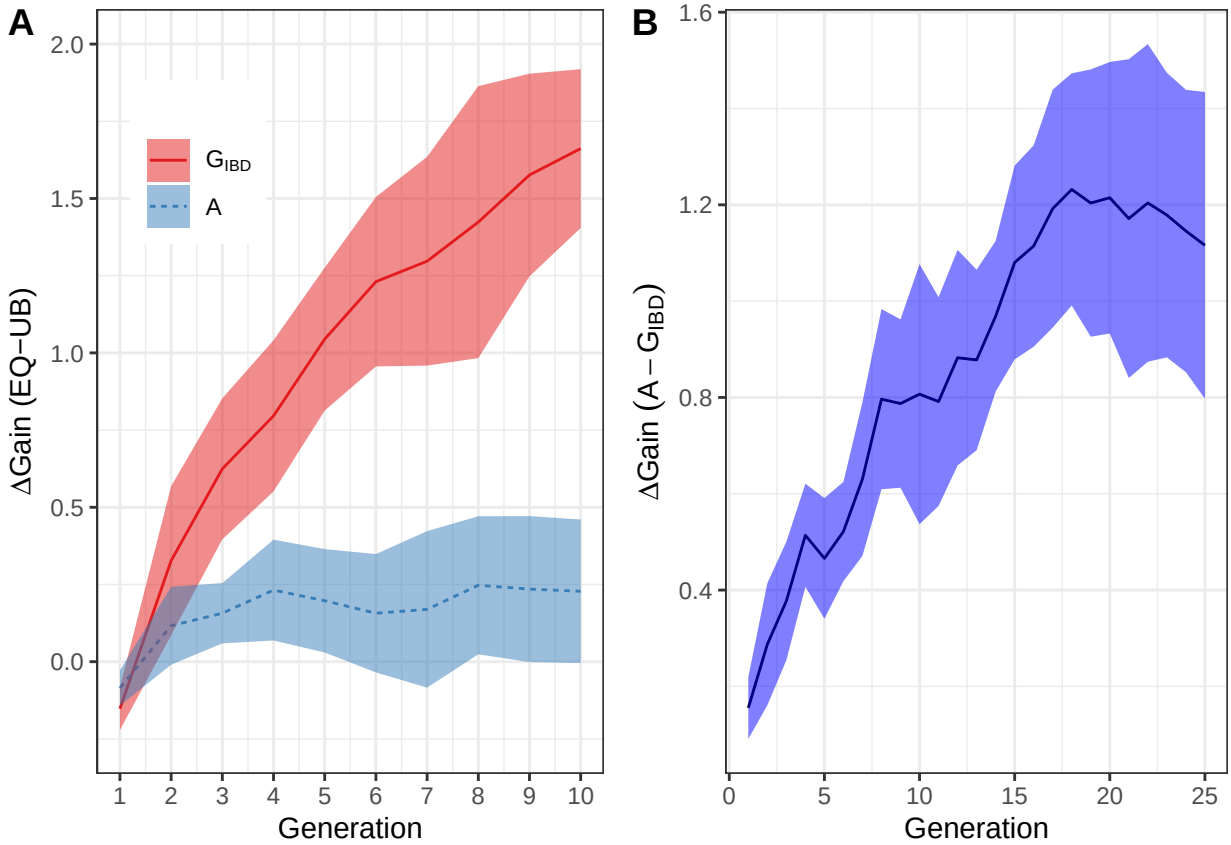

```
ggsave("Figure3.png",device="png",path="Figures",width=5,height=3,units="in")
```

Figure S2

```
data2 <- filter(data, ploidy=="2x",dF=="0.5%",method %in% c("OMA","OMA.UB"),
                kinship %in% c("A","G.IBD"))
data2$constraint <- ifelse(data2$method=="OMA.UB","UB","EQ")
data3a <- pivot_wider(data2[,c("replicate","gen","gain","kinship","constraint")],
                      names_from="constraint",values_from="gain")
data3a <- as.data.frame(data3a,check.names=F)
data3 <- data.frame(data3a,y=data3a$`EQ` - data3a$`UB`,check.names=F)

ans <- tapply(data3$y,list(data3$gen,data3$kinship),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat <- data.frame(x=rep(0:25,times=2), y=y,
                   ymin=CI[1,], ymax=CI[2,],
                   kinship=factor(rep(c("A","G.IBD"),each=26),levels=c("G.IBD","A"),ordered=T))

p1 <- ggplot(pdat) + geom_line(aes(x=x,y=y,col=kinship,linetype=kinship)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax,fill=kinship),alpha=0.5) +
  theme_bw() + xlab("Generation") + ylim(-0.25,2) +
  scale_linetype(name="",labels=c(expression(paste(G[IBD])),"A")) +
  scale_color_brewer(name="",palette="Set1",labels=c(expression(paste(G[IBD])),"A")) +
  scale_fill_brewer(name="",palette="Set1",labels=c(expression(paste(G[IBD])),"A")) +
  scale_x_continuous(name="Generation",limits=c(1,10),breaks=1:10,labels=1:10) +
  ylab(expression(paste(Delta,"Gain (EQ-UB)"))) +
```

```r
  theme(legend.position="inside",legend.position.inside = c(0.2,0.8))

data3a <- pivot_wider(data2[,c("replicate","gen","gain","kinship","constraint")],
                      names_from="kinship",values_from="gain")
data3a <- as.data.frame(data3a,check.names=F)
data3 <- data.frame(data3a,y=data3a$`A` - data3a$`G.IBD`,check.names=F)
data4 <- filter(data3,constraint=="EQ")
ans <- tapply(data4$y,data4$gen,t.test)
y <- sapply(ans,"[[",'estimate')
CI <- sapply(ans,"[[",'conf.int')
pdat <- data.frame(x=1:25, y=y[-1], ymin=CI[1,-1], ymax=CI[2,-1])

p2 <- ggplot(pdat) + geom_line(aes(x=x,y=y)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax),alpha=0.5,fill="blue") +
  theme_bw() + xlab("Generation") +
  ylab(expression(paste(Delta,"Gain ",(A-G[IBD]))))

data2 <- filter(data, ploidy=="2x",dF=="0.5%",method %in% c("OMA","OMA.UB"),
                kinship %in% c("A","G.IBD"))

ggarrange(p1,p2,nrow=1,common.legend = F,labels=c("A","B"))
```

```
## Warning: Removed 32 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

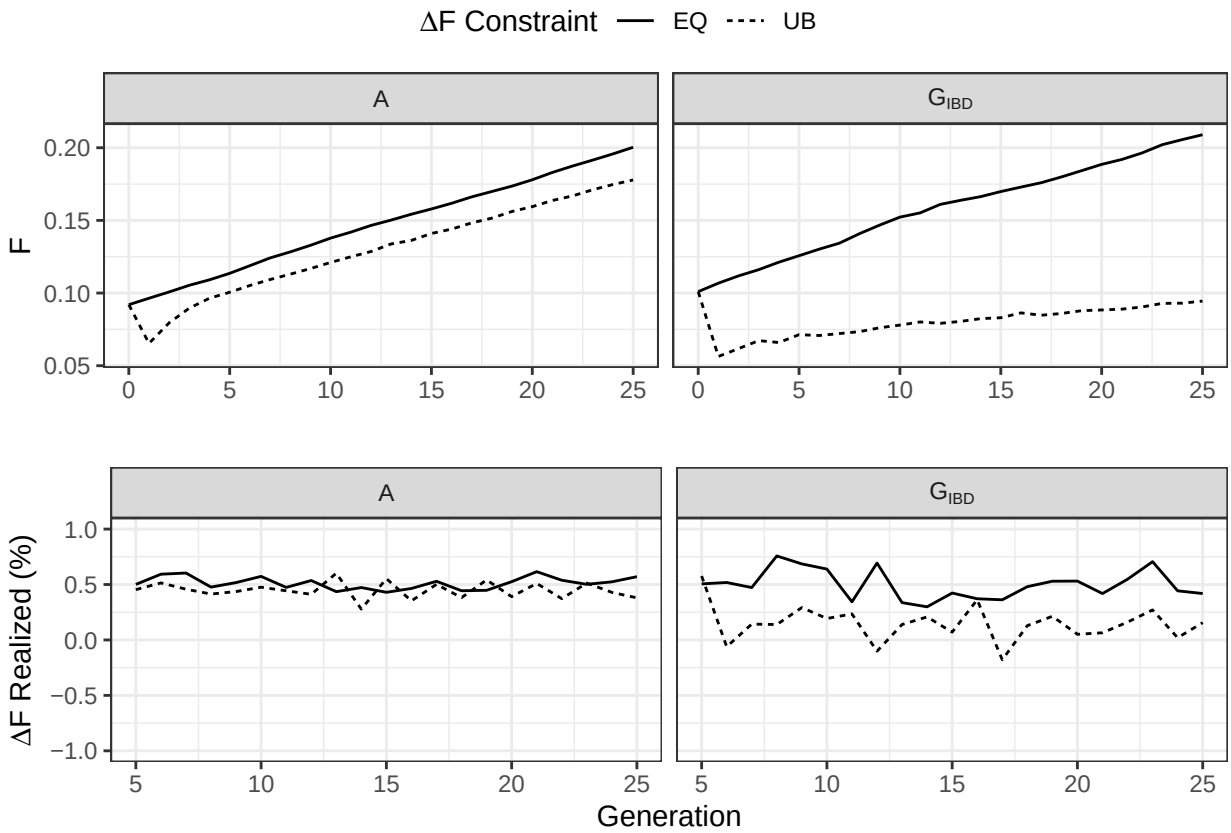


```r
ggsave("FigureS2.png",device="png",path="Figures",width=6,height=3,units="in")
```

Figure 4

```r
label.f <- function(x) {
  c("A",expression(paste(G[IBD])))
}

data2 <- filter(data, ploidy=="2x",dF=="0.5%",method %in% c("OMA","OMA.UB"),
                kinship %in% c("A","G.IBD"))
data2$constraint <- ifelse(data2$method=="OMA.UB","UB","EQ")
data3a <- aggregate(F.coeff~gen+constraint+kinship,data=data2,FUN=mean)
data3b <- aggregate(dFr~gen+constraint+kinship,data=data2,FUN=mean)
data3 <- merge(data3a,data3b,all=T)

p1 <- ggplot(data3,aes(x=gen,y=F.coeff,linetype=constraint)) +
  geom_line() + theme_bw() +
  facet_wrap(~kinship,nrow=1,labeller = as_labeller(label.f,default=label_parsed)) +
  ylab("F") + xlab("") +
  scale_linetype(name=expression(paste(Delta,"F Constraint")))

p2 <- ggplot(data3,aes(x=gen,y=dFr,linetype=constraint)) +
  geom_line() + theme_bw() +
  facet_wrap(~kinship,nrow=1,labeller = as_labeller(label.f,default=label_parsed)) +
  xlab("Generation") +
  scale_y_continuous(name=expression(paste(Delta,"F Realized (%)")),limits=c(-1,1)) +
  scale_linetype(name=expression(paste(Delta,"F Constraint"))) + xlim(5,25)

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

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
ggsave("Figure4.png",device="png",path="Figures",width=4,height=5,units="in")
```

Table S2

```
data2 <- filter(data, ploidy=="2x",dF=="0.5%",method %in% c("OMA","OMA.UB"),
                kinship %in% c("A","G.IBD"),gen==1)
data2$constraint <- ifelse(data2$method=="OMA.UB","UB","EQ")
alm <- lm(varA.genic~replicate+constraint+kinship+constraint:kinship,data2)
anova(alm)
```

```
## Analysis of Variance Table
##
## Response: varA.genic
##                    Df   Sum Sq   Mean Sq  F value    Pr(>F)
## replicate           9 0.100028 0.0111142 178.1043 < 2.2e-16 ***
## constraint          1 0.000037 0.0000366   0.5859 0.4506521
## kinship             1 0.001237 0.0012369  19.8219 0.0001326 ***
## constraint:kinship  1 0.000003 0.0000031   0.0499 0.8248739
## Residuals          27 0.001685 0.0000624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
alm <- lm(varA.diseq~replicate+constraint+kinship+constraint:kinship,data2)
anova(alm)
```

```
## Analysis of Variance Table
##
## Response: varA.diseq
```

```
##                     Df   Sum Sq   Mean Sq F value  Pr(>F)
## replicate            9 0.038636 0.0042929  2.4641 0.03381 *
## constraint           1 0.000060 0.0000600  0.0344 0.85415
## kinship              1 0.005367 0.0053671  3.0807 0.09057 .
## constraint:kinship   1 0.004026 0.0040261  2.3110 0.14009
## Residuals           27 0.047038 0.0017422
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
alm <- lm(varD.genic~replicate+constraint+kinship+constraint:kinship,data2)
anova(alm)
```

```
## Analysis of Variance Table
##
## Response: varD.genic
##                     Df   Sum Sq   Mean Sq  F value Pr(>F)
## replicate            9 0.035363 0.0039292 802.4201 <2e-16 ***
## constraint           1 0.000000 0.0000004   0.0725 0.7898
## kinship              1 0.000002 0.0000018   0.3600 0.5535
## constraint:kinship   1 0.000000 0.0000000   0.0030 0.9564
## Residuals           27 0.000132 0.0000049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

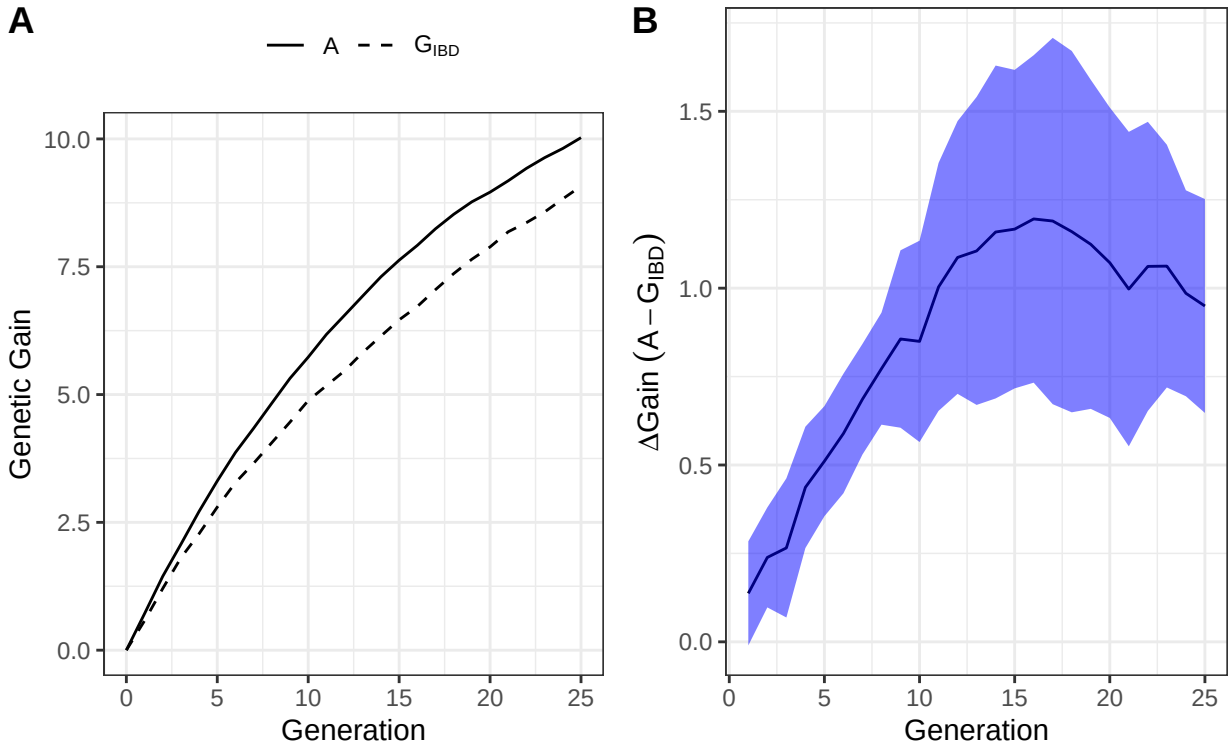
```
alm <- lm(varD.diseq~replicate+constraint+kinship+constraint:kinship,data2)
anova(alm)
```

```
## Analysis of Variance Table
##
## Response: varD.diseq
##                     Df   Sum Sq  Mean Sq F value    Pr(>F)
## replicate            9 0.021461 0.002385  2.7942   0.01866 *
## constraint           1 0.048232 0.048232 56.5174 4.361e-08 ***
## kinship              1 0.000706 0.000706  0.8273   0.37110
## constraint:kinship   1 0.006158 0.006158  7.2161   0.01221 *
## Residuals           27 0.023042 0.000853
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure S3

```
data2 <- filter(data, ploidy=="4x",method=="OMA",dF=="0.5%",kinship %in% c("A","G.IBD"))
data3 <- aggregate(gain~gen+kinship,data=data2,FUN=mean)
p1 <- ggplot(data3,aes(x=gen,y=gain,linetype=kinship)) +
  geom_line() + theme_bw() +
  scale_linetype_manual(name="",values=c(1,2),labels=c("A",expression(paste(G[IBD])))) +
  ylab("Genetic Gain") + theme(legend.position = "top") +
  xlab("Generation")

data3a <- pivot_wider(data2[,c("replicate","gen","gain","kinship")],
                      names_from="kinship",values_from="gain")
data3a <- as.data.frame(data3a,check.names=F)
data3 <- data.frame(data3a,y=data3a$`A` - data3a$`G.IBD`,check.names=F)
ans <- tapply(data3$y,data3$gen,t.test)
y <- sapply(ans,"[[",'estimate')
```

```
CI <- sapply(ans,"[[",'conf.int')
pdat <- data.frame(x=1:25, y=y[-1], ymin=CI[1,-1], ymax=CI[2,-1])

p2 <- ggplot(pdat) + geom_line(aes(x=x,y=y)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax),alpha=0.5,fill="blue") +
  theme_bw() + xlab("Generation") +
  ylab(expression(paste(Delta,"Gain ",(A-G[IBD]))))

ggarrange(p1,p2,nrow=1,labels=c("A","B"))
```



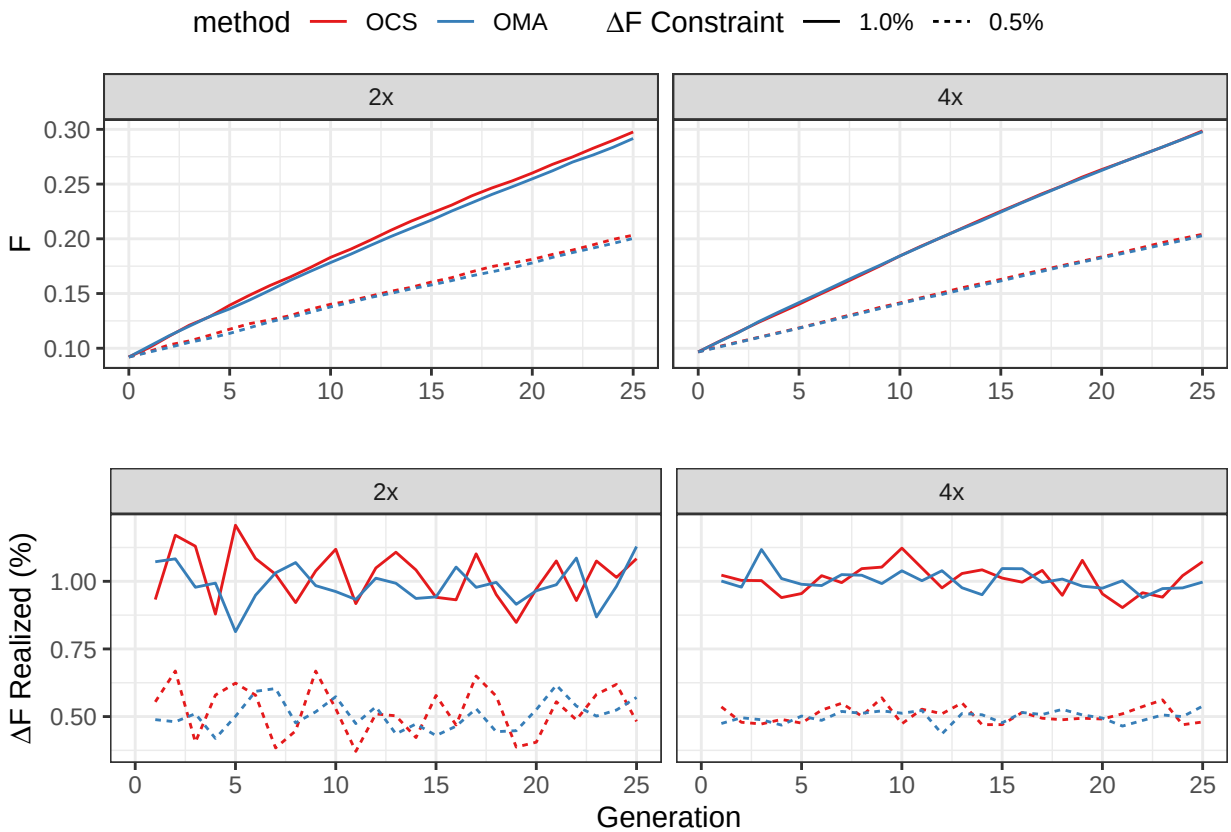

```
ggsave("FigureS3.png",device="png",path="Figures",width=6,height=3,units="in")
```

Figure S4

```
data2 <- filter(data, kinship=="A",method %in% c("OCS","OMA"))
data3a <- aggregate(F.coeff~gen+dF+ploidy+method,data=data2,FUN=mean)
data3b <- aggregate(dFr~gen+dF+ploidy+method,data=data2,FUN=mean)
data3 <- merge(data3a,data3b,all=T)

p1 <- ggplot(data3,aes(x=gen,y=F.coeff,linetype=dF,colour=method)) +
  geom_line() + theme_bw() +
  facet_wrap(~ploidy) + ylab("F") + xlab("") +
  scale_linetype(name=expression(paste(Delta,"F Constraint"))) +
  scale_colour_brewer(palette="Set1")

p2 <- ggplot(data3,aes(x=gen,y=dFr,linetype=dF,colour=method)) +
  geom_line() + theme_bw() +
  facet_wrap(~ploidy) + ylab(expression(paste(Delta,"F Realized (%)"))) +
  xlab("Generation") +
```

```
  scale_linetype(name=expression(paste(Delta,"F Constraint"))) +
  scale_colour_brewer(palette="Set1")

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

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
ggsave("FigureS4.png",device="png",path="Figures",width=5,height=5,units="in")
```

Figure 5

```
library(emmeans)
```

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

```
data2 <- filter(data, gen>=5,method %in% c("OMA","OCS"),kinship=="A")
data2$dF <- factor(data2$dF,ordered=T,levels=c("0.5%","1.0%"))
data3 <- aggregate(n.parent~replicate+dF+ploidy+method,data=data2,FUN=mean)

#panel A
plm <- lm(n.parent~replicate+method+dF+ploidy+method:dF,data3)
print(emmeans(plm,~dF:ploidy))
```

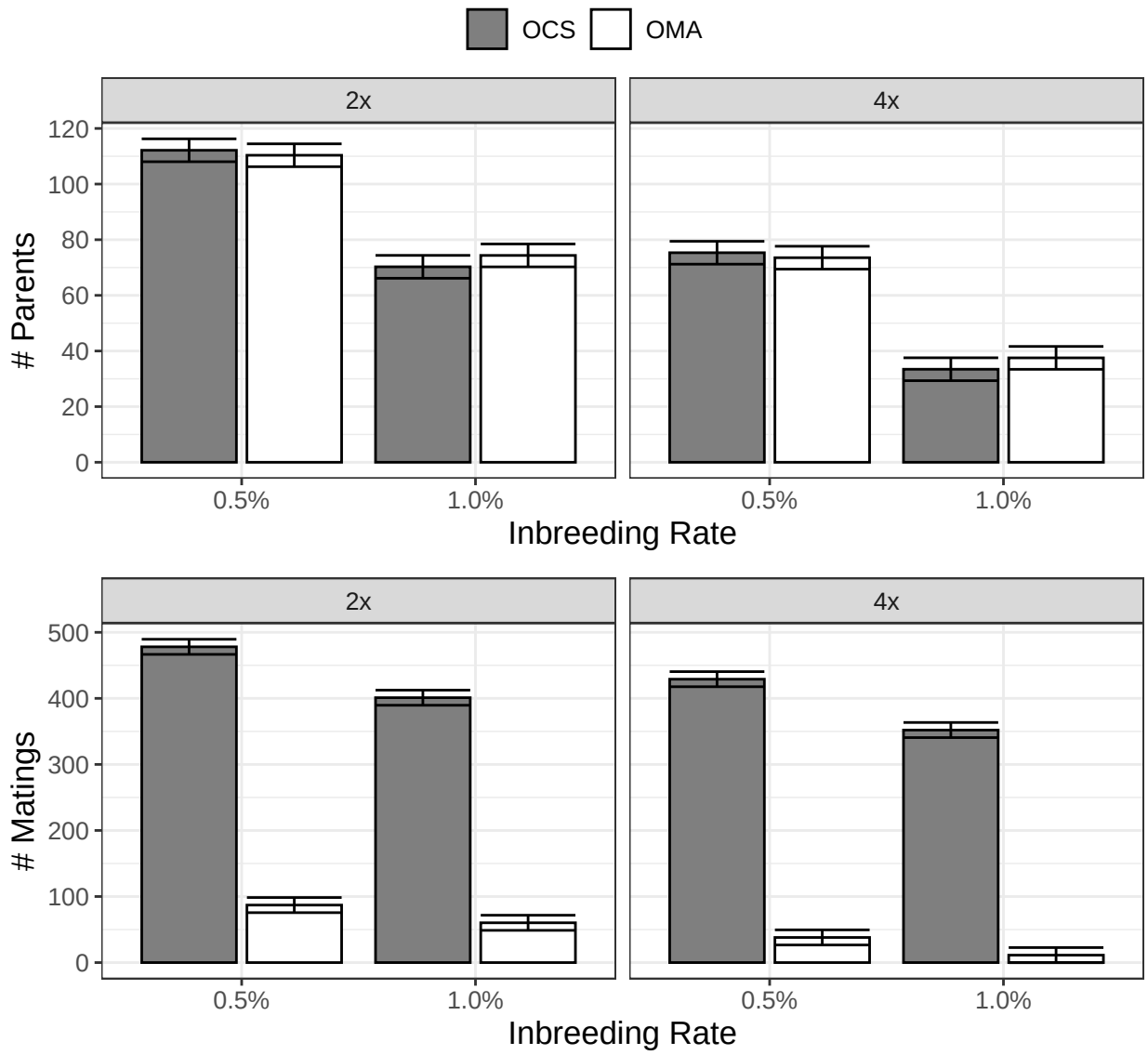
```
##  dF   ploidy emmean  SE df lower.CL upper.CL
##  0.5% 2x      111.2 1.6 66    108.1    114.4
##  1.0% 2x       72.3 1.6 66     69.1     75.5
##  0.5% 4x       74.4 1.6 66     71.2     77.6
```

```
##  1.0% 4x       35.5 1.6 66     32.3     38.7
##
## Results are averaged over the levels of: replicate, method
## Confidence level used: 0.95
```

```
pdat <- data.frame(emmeans(plm,~dF:ploidy:method))
p1 <- ggplot(pdat,aes(x=dF,y=emmean,ymin=lower.CL,ymax=upper.CL,fill=method)) +
  geom_col(position=position_dodge2(preserve="single"),colour="black") +
  geom_errorbar(position=position_dodge2(preserve="single")) +
  facet_wrap(~ploidy) + xlab("Inbreeding Rate") +
  scale_fill_manual(name="",values=c("grey50","white")) + theme_bw() +
  scale_y_continuous("# Parents",breaks=seq(0,140,by=20)) +
  theme(text=element_text(size=13))

#panel B
data3 <- aggregate(n.mate~replicate+dF+ploidy+method,data=data2,FUN=mean)
mlm <- lm(n.mate~replicate+method+dF+ploidy+method:dF,data3)
print(emmeans(mlm,~dF:ploidy:method))
```

```
##  dF   ploidy method emmean   SE df lower.CL upper.CL
##  0.5% 2x     OCS     478.0 5.73 66  466.596    489.5
##  1.0% 2x     OCS     401.0 5.73 66  389.594    412.5
##  0.5% 4x     OCS     429.0 5.73 66  417.601    440.5
##  1.0% 4x     OCS     352.0 5.73 66  340.599    363.5
##  0.5% 2x     OMA      87.0 5.73 66   75.587     98.5
##  1.0% 2x     OMA      60.2 5.73 66   48.789     71.7
##  0.5% 4x     OMA      38.0 5.73 66   26.591     49.5
##  1.0% 4x     OMA      11.2 5.73 66   -0.206     22.7
##
## Results are averaged over the levels of: replicate
## Confidence level used: 0.95
```

```
pdat <- data.frame(emmeans(mlm,~dF:ploidy:method))

p2 <- ggplot(pdat,aes(x=dF,y=emmean,ymin=lower.CL,ymax=upper.CL,fill=method)) +
  geom_col(position=position_dodge2(preserve="single"),colour="black") +
  geom_errorbar(position=position_dodge2(preserve="single")) +
  facet_wrap(~ploidy) + xlab("Inbreeding Rate") +
  scale_fill_manual(name="",values=c("grey50","white")) + theme_bw() +
  scale_y_continuous("# Matings") +
  theme(text=element_text(size=13))

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

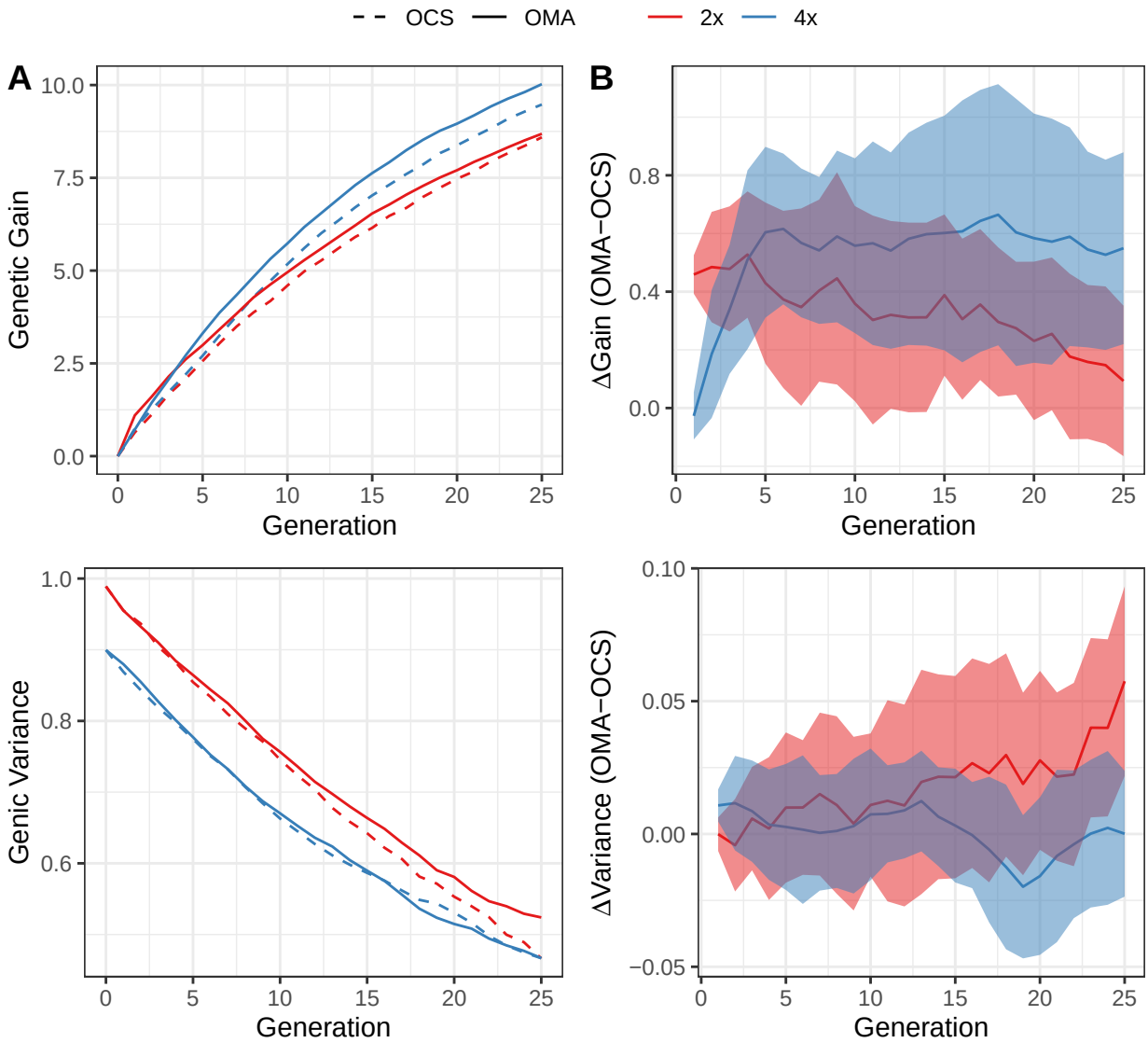



```
ggsave("Figure5.png",device="png",width=4,height=6,units="in",path="Figures")
```

Figure 6

```
## Compare OCS and OMA for kinship A
data2 <- filter(data, method %in% c("OCS","OMA"),kinship=="A",dF=="0.5%")
data3a <- aggregate(gain~gen+method+ploidy,data=data2,FUN=mean)
data3b <- aggregate(genicVar~gen+method+ploidy,data=data2,FUN=mean)
data3 <- merge(data3a,data3b,all=T)

p1 <- ggplot(data3,aes(x=gen,y=gain,linetype=method,colour=ploidy)) +
  geom_line() + theme_bw() +
  scale_linetype_manual(name="",values=c(2,1)) +
  scale_colour_brewer(palette="Set1",name="") +
  ylab("Genetic Gain") + xlab("") + theme(legend.position = "top") +
  xlab("Generation")
p2 <- ggplot(data3,aes(x=gen,y=genicVar,linetype=method,colour=ploidy)) +
```

```
  geom_line() + theme_bw() +
  scale_linetype_manual(name="",values=c(2,1)) +
  scale_colour_brewer(palette="Set1",name="") +
  ylab("Genic Variance") + xlab("") + theme(legend.position = "top") +
  xlab("Generation")

tmp <- pivot_wider(data2[,c("replicate","gen","gain","method","ploidy")],
                      names_from="method",values_from="gain")
data4a <- as.data.frame(tmp,check.names=F)
data4 <- data.frame(data4a,y=data4a$OMA - data4a$OCS,check.names=F)
ans <- tapply(data4$y,list(data4$gen,data4$ploidy),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat <- data.frame(gen=rep(0:25,times=2), y=y,
                   ymin=CI[1,], ymax=CI[2,],
                   ploidy=factor(rep(c("2x","4x"),each=26)))
pdat <- filter(pdat,gen>0)
p3 <- ggplot(pdat) + geom_line(aes(x=gen,y=y,col=ploidy)) +
  geom_ribbon(aes(x=gen,ymin=ymin,ymax=ymax,fill=ploidy),alpha=0.5) +
  theme_bw() + xlab("Generation") + scale_linetype(name="") +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") +
  scale_x_continuous(name="Generation") +
  ylab(expression(paste(Delta,"Gain (OMA-OCS)")))

tmp <- pivot_wider(data2[,c("replicate","gen","genicVar","method","ploidy")],
                      names_from="method",values_from="genicVar")
data4a <- as.data.frame(tmp,check.names=F)
data4 <- data.frame(data4a,y=data4a$OMA - data4a$OCS,check.names=F)
ans <- tapply(data4$y,list(data4$gen,data4$ploidy),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat <- data.frame(gen=rep(0:25,times=2), y=y,
                   ymin=CI[1,], ymax=CI[2,],
                   ploidy=factor(rep(c("2x","4x"),each=26)))
pdat <- filter(pdat,gen>0)
p4 <- ggplot(pdat) + geom_line(aes(x=gen,y=y,col=ploidy)) +
  geom_ribbon(aes(x=gen,ymin=ymin,ymax=ymax,fill=ploidy),alpha=0.5) +
  theme_bw() + xlab("Generation") + scale_linetype(name="") +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") +
  scale_x_continuous(name="Generation") +
  ylab(expression(paste(Delta,"Variance (OMA-OCS)")))

ggarrange(p1,p3,p2,p4,nrow=2,ncol=2,common.legend = T,labels=c("A","B","",""))
```

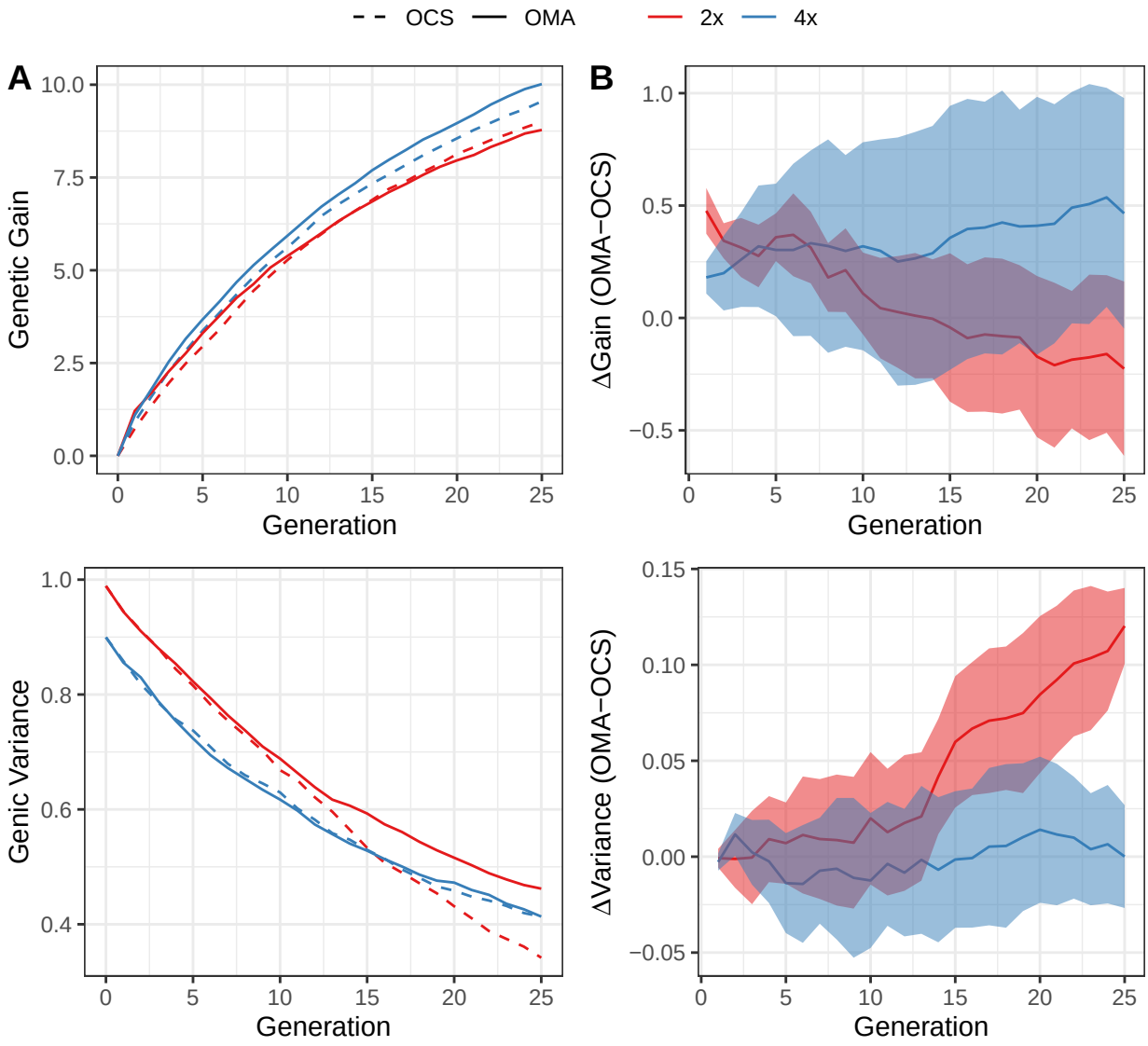



```
ggsave("Figure6.png",device="png",height=6,width=6,units="in",path="Figures")
```

Figure S5 1% inbreeding

```
data2 <- filter(data,method %in% c("OCS","OMA"), kinship=="A", dF=="1.0%")
data3a <- aggregate(gain~gen+method+ploidy,data=data2,FUN=mean)
data3b <- aggregate(genicVar~gen+method+ploidy,data=data2,FUN=mean)
data3 <- merge(data3a,data3b,all=T)

p1 <- ggplot(data3,aes(x=gen,y=gain,linetype=method,colour=ploidy)) +
  geom_line() + theme_bw() +
  scale_linetype_manual(name="",values=c(2,1)) +
  scale_colour_brewer(palette="Set1",name="") +
  ylab("Genetic Gain") + xlab("") + theme(legend.position = "top") +
  xlab("Generation")
p2 <- ggplot(data3,aes(x=gen,y=genicVar,linetype=method,colour=ploidy)) +
  geom_line() + theme_bw() +
  scale_linetype_manual(name="",values=c(2,1)) +
```

```
  scale_colour_brewer(palette="Set1",name="") +
  ylab("Genic Variance") + xlab("") + theme(legend.position = "top") +
  xlab("Generation")

tmp <- pivot_wider(data2[,c("replicate","gen","gain","method","ploidy")],
                     names_from="method",values_from="gain")
data4a <- as.data.frame(tmp,check.names=F)
data4 <- data.frame(data4a,y=data4a$OMA - data4a$OCS,check.names=F)
ans <- tapply(data4$y,list(data4$gen,data4$ploidy),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat <- data.frame(gen=rep(0:25,times=2), y=y,
                   ymin=CI[1,], ymax=CI[2,],
                   ploidy=factor(rep(c("2x","4x"),each=26)))
pdat <- filter(pdat,gen>0)
p3 <- ggplot(pdat) + geom_line(aes(x=gen,y=y,col=ploidy)) +
  geom_ribbon(aes(x=gen,ymin=ymin,ymax=ymax,fill=ploidy),alpha=0.5) +
  theme_bw() + xlab("Generation") + scale_linetype(name="") +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") +
  scale_x_continuous(name="Generation") +
  ylab(expression(paste(Delta,"Gain (OMA-OCS)")))

tmp <- pivot_wider(data2[,c("replicate","gen","genicVar","method","ploidy")],
                     names_from="method",values_from="genicVar")
data4a <- as.data.frame(tmp,check.names=F)
data4 <- data.frame(data4a,y=data4a$OMA - data4a$OCS,check.names=F)
ans <- tapply(data4$y,list(data4$gen,data4$ploidy),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat <- data.frame(gen=rep(0:25,times=2), y=y,
                   ymin=CI[1,], ymax=CI[2,],
                   ploidy=factor(rep(c("2x","4x"),each=26)))
pdat <- filter(pdat,gen>0)
p4 <- ggplot(pdat) + geom_line(aes(x=gen,y=y,col=ploidy)) +
  geom_ribbon(aes(x=gen,ymin=ymin,ymax=ymax,fill=ploidy),alpha=0.5) +
  theme_bw() + xlab("Generation") + scale_linetype(name="") +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") +
  scale_x_continuous(name="Generation") +
  ylab(expression(paste(Delta,"Variance (OMA-OCS)")))

ggarrange(p1,p3,p2,p4,nrow=2,ncol=2,common.legend = T,labels=c("A","B","",""))
```

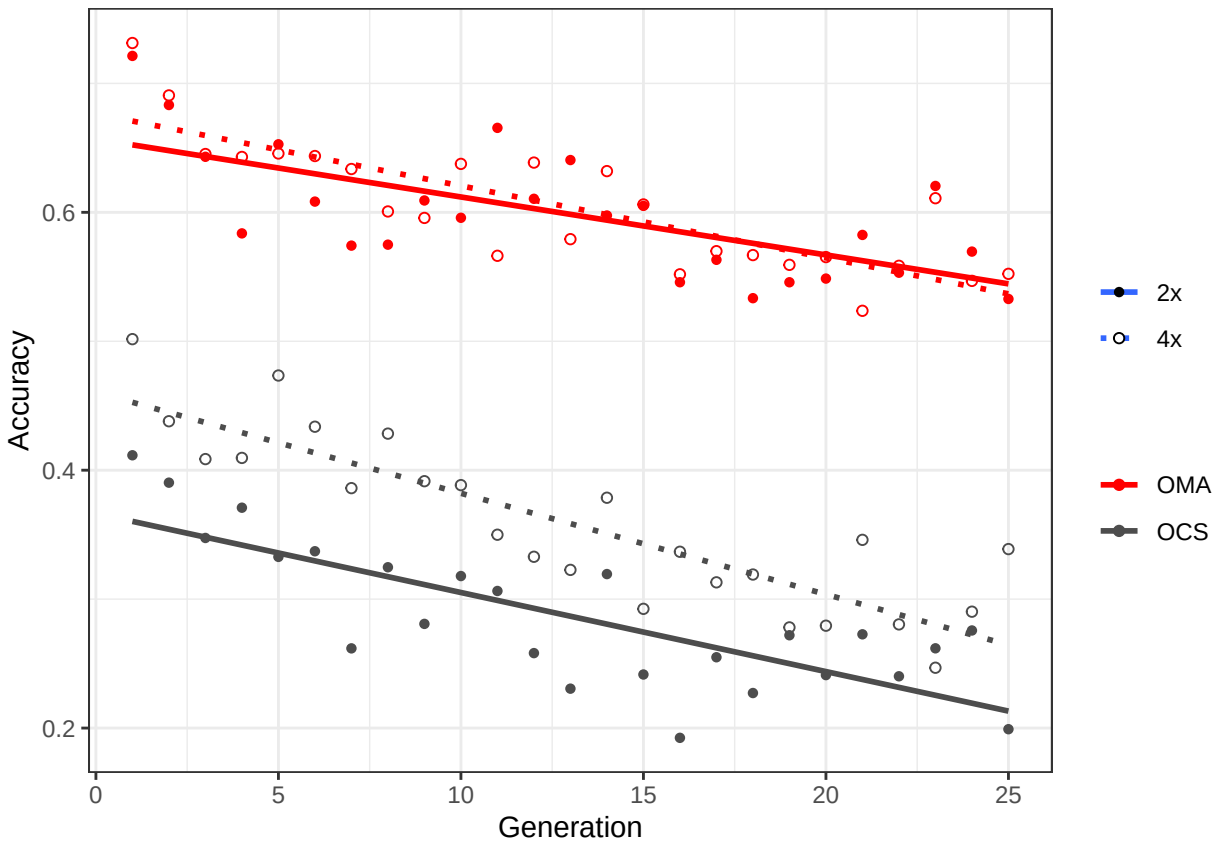



```
ggsave("FigureS5.png",device="png",height=6,width=6,units="in",path="Figures")
```

Figure 7

```
data2 <- filter(data, method %in% c("OCS","OMA"),kinship=="A",dF=="0.5%",gen>0)
data3 <- aggregate(accuracy~gen+method+ploidy,data=data2,FUN=mean)
data3$method <- factor(data3$method,levels=c("OMA","OCS"),ordered=T)
ggplot(data3,aes(x=gen,y=accuracy,colour=method,shape=ploidy,linetype=ploidy)) +
  stat_smooth(method="lm",se=F) +
  geom_point() +
  scale_colour_manual(name="",values=c("red","grey30")) +
  scale_shape_manual("",values=c(16,1)) +
  scale_linetype_manual("",values=c(1,3)) +
  theme_bw() + xlab("Generation") +
  scale_y_continuous("Accuracy")
```

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



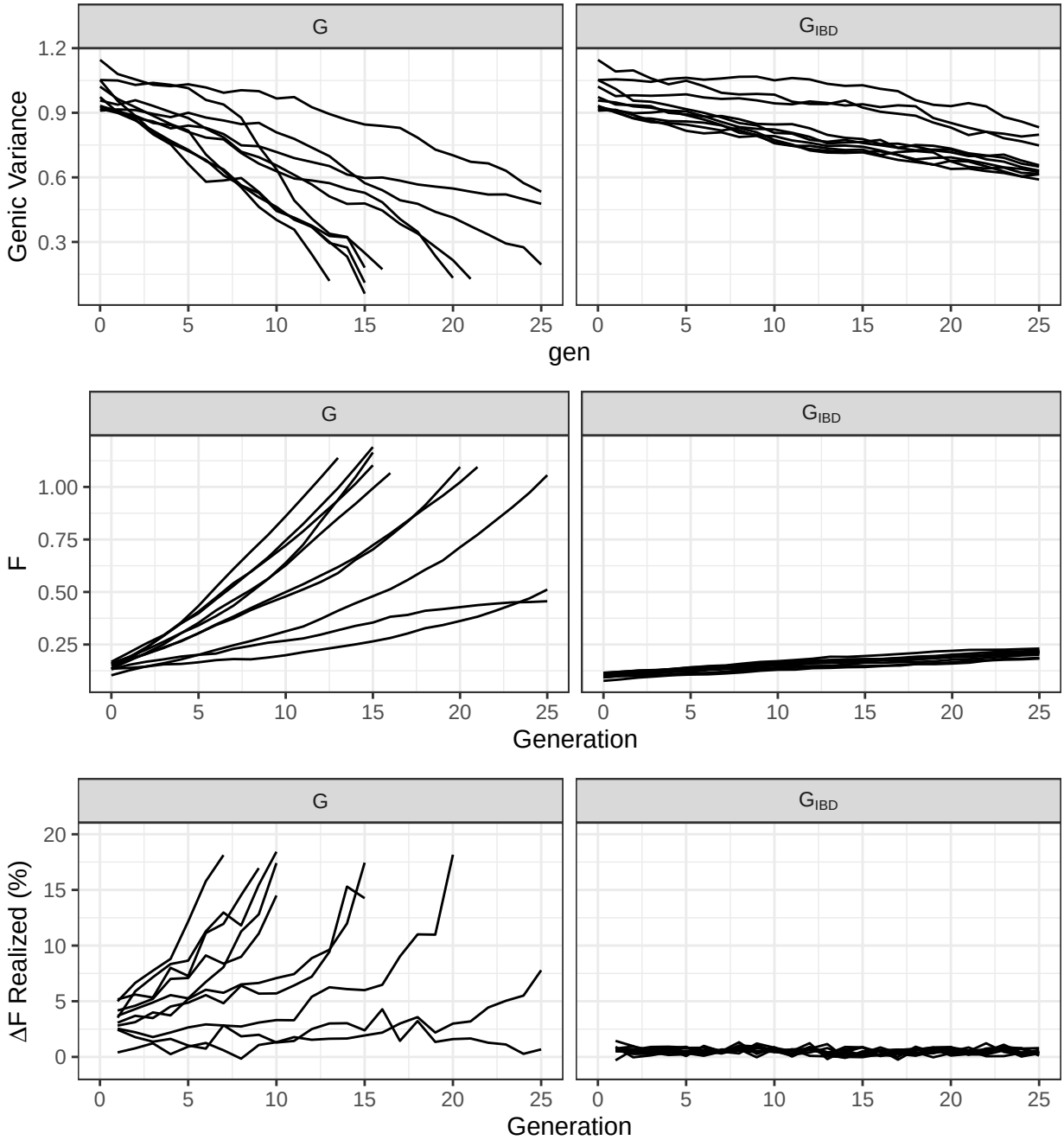

```
ggsave("Figure7.png",device="png",height=3,width=4,units="in",path="Figures")
```

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

Figures S6 G matrix

```
label.f <- function(x) {
  c("G",expression(paste(G[IBD])))
}

data2 <- filter(data, ploidy=="2x",dF=="0.5%",kinship!="A",method=="OMA")

p1 <- ggplot(data2,aes(x=gen,y=genicVar,group=replicate)) +
  geom_line() + theme_bw() + ylab("Genic Variance") +
  facet_wrap(~kinship,nrow=1,labeller = as_labeller(label.f,default=label_parsed))

p2 <- ggplot(data2,aes(x=gen,y=F.coeff,group=replicate)) +
  geom_line() + theme_bw() + xlab("Generation") + ylab("F") +
  facet_wrap(~kinship,nrow=1,labeller = as_labeller(label.f,default=label_parsed))

p3 <- ggplot(data2,aes(x=gen,y=dFr,group=replicate)) +
  geom_line() + theme_bw() + xlab("Generation") +
  scale_y_continuous(name=expression(paste(Delta,"F Realized (%)")),limits=c(-1,20)) +
  facet_wrap(~kinship,nrow=1,labeller = as_labeller(label.f,default=label_parsed))

ggarrange(p1,p2,p3,nrow=3,common.legend=T)
```

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 10 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
ggsave("FigureS6.png",device="png",path="Figures",width=5,height=7,units="in")
```