Selecting Chromosomes for Polygenic Traits: Algorithms and Complexity

Or Zuk
Department of Statistics and Data Science, The Hebrew University of Jerusalem, Israel

Abstract

We define and study the problem of *genomic block selection* for multiple complex traits. In this problem, one constructs a genome by selecting different genomic parts (e.g. chromosomes) from different source genomes. The constructed genome is associated with a vector of polygenic scores, obtained by summing the polygenic scores of the different genomic parts, and the goal is to minimize a given loss function of this vector. The problem is motivated by several emerging technologies: chromosome substitution lines in crop breeding, where chromosomal segments from wild relatives are combined to improve polygenic traits such as yield and stress tolerance; chromosome transfer between yeast strains for optimizing complex industrial phenotypes; and chromosomal transplantation technologies in mammalian cells. We suggest and study several natural loss functions relevant for both quantitative and threshold traits, and show that the problem is NP-complete even for a single trait and two copies, yet only weakly so, being pseudo-polynomially solvable for any fixed number of traits. We propose three algorithms with complementary roles: a Branch-and-Bound algorithm that returns the certified global optimum for any monotone loss, a fast Block-Coordinate-Descent (BCD) heuristic with random restarts that applies to any loss, and a semidefinite-programming (SDP) relaxation that provides a certified lower bound on the optimal loss for quadratic losses, and hence an optimality-gap bound when paired with the BCD solution - empirically tight in our experiments. Using the infinitesimal model for genetic architecture, we further derive, for linear losses, a closed-form approximation for the expected gain of block selection relative to random selection across multiple traits. On yeast-scale simulations BCD matches the certified Branch-and-Bound optimum on 100% of threshold-loss instances at 466× the speed, attains a certified optimality gap of at most ≈10% of the SDP lower bound for stabilizing-loss instances, and the realized gain roughly matches the analytic prediction.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Applied computing → Computational genomics; Applied computing → Computational genomics

Keywords and phrases polygenic scores, combinatorial optimization, genomic block selection, NP-hardness, semidefinite programming, synthetic genomics

Digital Object Identifier 10.4230/LIPIcs.WABI.2026.28

Supplementary Material *Software*: `https://github.com/orzuk/EmbryoSelectionCalculator`

Funding Supported by the Israeli Science Foundation (grants #2830/20 and #2392/22).

1 Introduction

In several breeding and biotechnology contexts, it is possible or becoming feasible to combine genomic parts from different source individuals into a single genome. In plant breeding, chromosome substitution lines are routinely created by introgressing chromosomal segments from wild relatives into elite cultivars, for example barley chromosome arms into wheat [16,33], in order to improve polygenic traits such as yield, disease resistance and stress tolerance. In yeast, the completion of the Sc2.0 synthetic genome project [34] and tools for chromosome transfer between strains [31, 38] open the possibility of constructing chimeric genomes optimized for complex industrial phenotypes such as ethanol tolerance and thermotolerance, which are controlled by dozens to hundreds of quantitative trait loci (QTL) [3, 4]. In

mammalian cells, high-fidelity chromosomal transplantation has been demonstrated [29, 30, 32], and programmable recombination systems now enable targeted, large-scale DNA rearrangements [12].

For highly polygenic traits where no individual gene has a large effect, selection of individual chromosomes or large-scale genomic regions from available genomes may complement local gene-editing approaches using the CRISPR-CAS9 system [21] that affect only one or a handful of genes. Chromosomal selection leverages existing genetic material without introducing new mutations, but comes with a trade-off: entire chromosomes carry both favorable and unfavorable alleles (linkage drag), whereas targeted editing can modify specific loci with precision.

Choosing which blocks to combine requires a quantitative predictor of each assembled genome's phenotype. Polygenic scores (PS) provide this, combining the contributions of multiple genomic alleles with coefficients typically fitted from Genome-Wide-Association-Studies (GWAS). GWAS originated for human complex traits [23, 25, 40] and has since been adopted in livestock [27], crops [11], and yeast [3, 31]; accuracy continues to improve with growing sample sizes and advances in statistical methods [9, 18, 28].

Here we formulate and study the *genomic block selection* problem for multiple quantitative traits. In this problem, multiple copies are available for each chromosome (or possibly a smaller genomic part), and based on these copies' PS we select one of them. These choices yield a genome assembled from the different chromosomal parts. We address the following two main questions:

1. How should the chromosomes be selected in order to maximize utility across multiple traits? When selecting for T traits, each selection $\mathbf{c}$ of chromosomes leads to an overall genomic score vector $\boldsymbol{X}_{\mathbf{c}} \in \mathbb{R}^T$. A loss function $\mathcal{L} : \mathbb{R}^T \to \mathbb{R}$ is defined and our goal is to find the selection $\mathbf{c}$ minimizing the loss $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$, and compare it to the loss obtained for random selection. When C copies are available for each of the M chromosomes, the total number of possible selections C^M is exponential in M, hence the need to design efficient general algorithms for the problem.
2. What is the expected gain when using optimal chromosomal selection? How does it compare to the baseline, i.e. selecting genomes at random, as well as to the whole-genome selection procedure in which one of the C source genomes is selected as a whole [22, 26]?

Our main contributions are threefold. First, we formulate the genomic block selection problem mathematically and establish its weak NP-hardness even in simple cases: a single-trait stabilizing loss and a two-trait all-pass loss, each with two copies. Second, we develop three algorithms with complementary theoretical and practical roles: (i) a Branch-and-Bound (B&B) algorithm that returns the certified global optimum for monotone losses, in which the number of Pareto-non-dominated partial selections retained at each stage empirically grows much slower than exhaustive search over the regimes we tested; (ii) a simple Block-Coordinate-Descent (BCD) heuristic with random restarts, applicable to any loss, utilizing the fact that updating a single block's selection touches only T coordinates of the score vector and therefore runs in $O(M\,C\,T)$ per pass over all blocks; and (iii) a semidefinite-programming (SDP) relaxation that provides a certified lower bound on the loss for quadratic losses (empirically tight in our simulations), so it certifies the optimality gap of the BCD solution. Third, we estimate the expected gain achieved by chromosomal selection, deriving an analytic approximation for linear losses and validating the algorithms on a yeast-like simulation scenario.

2 The Chromosomal Selection Problem

2.1 Background and Selection Problems for Polygenic Scores

Consider a genome composed of M distinct genomic blocks, typically whole chromosomes. For haploid organisms (e.g. haploid yeast with $M = 16$), each block corresponds to a single chromosome. Diploid organisms carry two homologs per chromosome, so a block may be a chromosome type (M) or an individual homolog ($2M$); a diploid yeast strain, for instance, has 16 chromosome types and 32 homologs. For polyploid organisms one may further distinguish sub-genomes and homologs. For example, hexaploid wheat has 7 chromosome groups across 3 homeologous sub-genomes, so M can range from 7 (one block per chromosome group) through 21 (7×3, one block per sub-genomic chromosome) to 42 ($7 \times 3 \times 2$, one block per homolog), depending on the resolution of selection. We associate with each block a polygenic-score vector representing the genetic contribution of the block to T traits of interest. Suppose that we have C distinct source genomes, hence C copies are available for each block overall. Our goal is to select one copy for each block, possibly under constraints, yielding a full genome with desired properties in terms of the resulting polygenic score. Two natural selection schemes are: (i.) *whole-genome selection*, where one of the C source genomes is chosen, so all selected blocks belong to the same genome; (ii.) *chromosomal (block) selection*, where different blocks can be drawn from different sources, yielding a chimeric genome and a search space of size C^M. Concrete instances include yeast strain panels (e.g. C yeast strains, each with $M = 16$ chromosomes, give C^{16} possible chimeras) and wheat breeding (C donor lines, including wild relatives, provide chromosomal segments for each of the M chromosome groups, and one seeks the best combination of segments to optimize multiple agronomic traits). Figure 1(a) illustrates whole-genome selection, in which a single source genome is chosen, while Figure 1(b) illustrates chromosomal selection from multiple source genomes.

2.2 Preliminaries

For a natural number $n \in \mathbb{N}$ denote the set $\{1, .., n\}$ by $[n]$. For two natural numbers $m \leq n$ denote the set $\{m, m+1, .., n\}$ by $[m, n]$. The vector of all zeros (ones) of length n is denoted $\mathbf{0}_n$ ($\mathbf{1}_n$). For a polytope $\Delta \in \mathbb{R}^n$, define the projection operator $\mathbb{P}_\Delta(x) \equiv \arg\min_{y \in \Delta} \|y - x\|^2$.

Tensors. Let $\boldsymbol{X} \in \mathbb{R}_{m \times n \times p}$ be a 3^{rd}-order tensor with elements X_{ijk}. We denote lower-dimensional fibers of a tensor using the $\bullet$ subscript. For example, $\boldsymbol{X}_{ij\bullet}$ denotes the vector $(X_{ij1}, .., X_{ijp}) \in \mathbb{R}^p$. Similarly, $\boldsymbol{X}_{\bullet j\bullet}$ is a matrix of size $m \times p$ containing all elements $X_{ijk}, \forall i \in [m], \forall k \in [p]$. We also describe sub-tensors obtained by taking subsets of the indices across each dimension. For example, $\boldsymbol{X}_{[i][j]\bullet}$ is a 3^{rd}-order tensor of size $i \times j \times p$ obtained by taking the first i and j coordinates on the first and second dimension, respectively.

For a 3^{rd}-order tensor $\boldsymbol{X} \in \mathbb{R}_{m \times n \times p}$ and a matrix $\mathbf{A} \in \mathbb{R}_{m \times n}$, define the *2-mode* tensor-by-matrix product [1, 24], as a matrix $[\boldsymbol{X} \times_2 \mathbf{A}] \in \mathbb{R}_{m \times p}$, with elements defined by:

$$[\boldsymbol{X} \times_2 \mathbf{A}]_{ik} = \sum_{j=1}^{n} X_{ijk} A_{ij}, \ \forall i \in [m], \forall k \in [p]. \tag{1}$$

Gaussian distribution. For the multivariate Gaussian distribution with mean μ and variance Σ, denote by $\phi(\mathbf{x}; \mu, \Sigma)$ and $\Phi(\mathbf{x}; \mu, \Sigma)$ the density function and cumulative distribution function, respectively. When μ, Σ are omitted, Φ and ϕ refer to the standard multivariate

Gaussian distribution with mean zero and identity covariance matrix. For a measurable set $\mathcal{A} \subset \mathbb{R}^T$ denote by $\Phi(\mathcal{A}; \mu, \Sigma)$ the probability of the set under the Gaussian distribution, i.e. $\Phi(\mathcal{A}; \mu, \Sigma) \equiv \int_{\mathcal{A}} \phi(\mathbf{x}; \mu, \Sigma) d\mathbf{x}$.

2.3 Chromosomal Selection for Multiple Traits

Let $\boldsymbol{X} \in \mathbb{R}_{M \times C \times T}$ be a 3^{rd}-order tensor of polygenic scores, where X_{ijk} denotes the score in chromosome i of copy j for trait k. Let $\mathbf{c} = (c_1, .., c_M) \in [C]^M$ be a selection vector. The associated selected polygenic vector is defined as $\boldsymbol{X}_{\mathbf{c}} \equiv \sum_{i=1}^{M} X_{ic_i\bullet} \in \mathbb{R}^T$, with $\boldsymbol{X}_{\mathbf{c}_k}$ denoting its k-th element $(\boldsymbol{X}_{\mathbf{c}})_k$, $\forall k \in [T]$. Our goal is to find the selected score vector $\boldsymbol{X}_{\mathbf{c}}$ minimizing a loss function of our choice. The *multi-trait chromosomal selection problem* is defined as follows:

▶ **Problem 1.** *Given a 3^{rd}-order tensor of scores $\boldsymbol{X} \in \mathbb{R}_{M \times C \times T}$, and a loss function $\mathcal{L} : \mathbb{R}^T \to \mathbb{R}$, find a vector $\mathbf{c}^* \in [C]^M$ minimizing the loss: $\mathbf{c}^* \equiv \underset{\mathbf{c} \in [C]^M}{argmin} \mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$.*

Table 1 lists several examples of natural loss functions of interest for both quantitative and threshold traits. The computational difficulty of Problem 1 above depends on the loss function $\mathcal{L}$. For example, when $\mathcal{L}$ is linear in the polygenic score vector $\boldsymbol{X}_{\mathbf{c}}$, we can select the optimal copy for each block independently, and the computational problem becomes trivial; this is the case when maximizing a weighted combination of quantitative traits. For non-linear losses (e.g. minimizing the overall failure probability across multiple threshold traits), selecting the best chromosome may be computationally challenging since the scores of all chromosomes must be considered jointly. In Section 3, we propose three algorithms for finding the optimal selection, applicable to general classes of loss functions.

Many natural loss functions are *monotone* in the scores vector, a property we exploit in some of our algorithms. Denote by $\mathbf{x} \preceq \mathbf{y}$ the coordinatewise product order on $\mathbb{R}^n$ (i.e. $x_i \leq y_i$ for all i), and write $\mathbf{x} \prec \mathbf{y}$ if $\mathbf{x} \preceq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$. A loss $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ is *monotonically non-increasing* if $\mathbf{x} \preceq \mathbf{y}$ implies $\mathcal{L}(\mathbf{x}) \geq \mathcal{L}(\mathbf{y})$.

Table 1 Example loss functions $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$ and their properties. In all rows $\boldsymbol{X}_{\mathbf{c}} \in \mathbb{R}^T$ is the selected score vector and $\boldsymbol{X}_{\mathbf{c}_k}$ its k-th coordinate; $\mathbf{w} \in \mathbb{R}^T$ is a trait-weight vector. (i) The linear loss applies to quantitative traits (e.g. growth rate, stress tolerance, metabolite yield). (ii) The stabilizing loss [20, 35, 36] penalizes deviations of the resulting score from the trait optimum (set to zero w.l.o.g.); this is appropriate when a trait is already near its optimum and large deviations are harmful. (iii)–(iv) For T threshold traits (e.g. disease susceptibility or failure to meet a quality standard), D_k is a binary random variable (the status of trait k) whose distribution depends on the selected score $\boldsymbol{X}_{\mathbf{c}}$, and $\mathbf{D} = (D_1, \ldots, D_T)$; the crossing probability $P(D_k = 1 \mid \boldsymbol{X}_{\mathbf{c}})$ follows a liability-threshold model [14] and depends on the trait's prevalence, heritability, and the cross-trait residual covariance (full model in Appendix B). The threshold-linear loss minimizes a weighted sum of crossing probabilities; the all-pass loss takes the negative probability that no threshold is crossed.

Loss	Formula	Monotone?	Convex?	Algorithms
Linear	$-\sum_{k=1}^{T} w_k \boldsymbol{X}_{\mathbf{c}_k}$, $\mathbf{w} \geq 0$	YES	YES	Simple Selection
Stabilizing	$\sum_{k=1}^{T} w_k \boldsymbol{X}_{\mathbf{c}_k}^2$	NO	YES	BCD; SDP (bound)
Threshold-linear	$\sum_{k=1}^{T} w_k P(D_k = 1 \mid \boldsymbol{X}_{\mathbf{c}})$	YES	NO	B&B (exact) / BCD
All-pass	$-P(\mathbf{D} = \mathbf{0}_T \mid \boldsymbol{X}_{\mathbf{c}})$	YES	NO	B&B (exact) / BCD

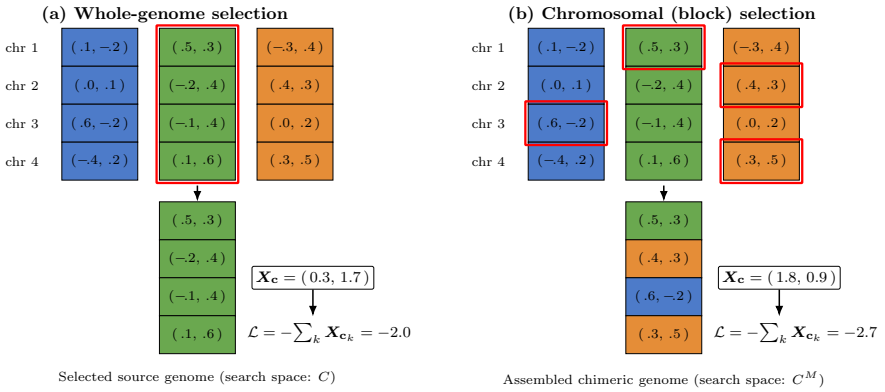

Figure 1 Two paradigms for selecting a genome from C source genomes, each providing one copy of M chromosomes (or genomic blocks). The example shown uses $C = 3$ source genomes (blue, green, orange), $M = 4$ chromosomes, and $T = 2$ traits, with each cell labelled by its vector of polygenic scores $\boldsymbol{X}_{ij\bullet} \in \mathbb{R}^2$. The two paradigms differ in which subset is selected (red boxes). **(a)** Whole-genome selection picks the single source whose aggregated polygenic-score vector $\sum_i \boldsymbol{X}_{ij\bullet}$ is optimal; the red box wraps around the selected genome shown as an entire column (Source 2), so the output is a monochromatic genome with $\boldsymbol{X}_{\mathbf{c}} = (0.3, 1.7)$. **(b)** Chromosomal (block) selection assigns, jointly across chromosomes, a source $c_i \in [C]$ to each chromosome i, giving a chimeric genome $\boldsymbol{X}_{\mathbf{c}} = \sum_i \boldsymbol{X}_{ic_i\bullet}$ that minimizes the chosen loss $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$; a red box wraps around each selected block. Here the loss is the linear loss $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) = -\sum_k \boldsymbol{X}_{\mathbf{c}_k}$ with equal weights, i.e. maximizing the total score across both traits ($1.8 + 0.9 = 2.7$ versus $0.3 + 1.7 = 2.0$ for whole-genome selection). The output is a mosaic whose blocks are colored by their source of origin, with $\boldsymbol{X}_{\mathbf{c}} = (1.8, 0.9)$. Sources may represent, e.g. different yeast strains, crop donor lines, or engineered mammalian cells via chromosome transplantation.

2.3.1 The Gain due to Selection

▶ **Definition 2.** *For a tensor $\boldsymbol{X}$ of scores and a loss function $\mathcal{L}$, let $\mathbf{c} \sim \mathrm{Unif}([C]^M)$ be a uniformly random selection (each c_i drawn i.i.d. uniformly from $[C]$), so that $\mathbb{E}_{\mathbf{c}}\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) = C^{-M}\sum_{\mathbf{c}\in[C]^M}\mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$. We define the* Gain G *due to chromosomal selection and the* Gain G_e *due to whole-genome selection as the difference between this random-selection baseline and the optimal loss under each scheme:*

$$G(\boldsymbol{X};\mathcal{L}) \equiv \mathbb{E}_{\mathbf{c}}\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) - \mathcal{L}(\boldsymbol{X}_{\mathbf{c}^*})$$
$$G_e(\boldsymbol{X};\mathcal{L}) \equiv \mathbb{E}_{\mathbf{c}}\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) - \min_{j\in[C]} \mathcal{L}\Big(\sum_{i=1}^{M} \boldsymbol{X}_{ij\bullet}\Big). \tag{2}$$

The gain $G_e(\boldsymbol{X};\mathcal{L})$ is similar in spirit to previous definitions given in [22, 26] for human embryo selection, but generalized here for a general loss whereas [22, 26] defined the gain only for additive losses for quantitative and disease traits. By definition $G_e(\boldsymbol{X};\mathcal{L}) \le G(\boldsymbol{X};\mathcal{L})$; both are functions of the score tensor $\boldsymbol{X}$. Averaging over $\boldsymbol{X}$ drawn from a statistical model yields the *expected gains* $\mathbb{E}_{\boldsymbol{X}}G$ and $\mathbb{E}_{\boldsymbol{X}}G_e$. In Section 4 we derive closed-form approximations of the expected gain for linear loss functions.

3 Algorithms and Complexity

The tractability of Problem 1 depends on the loss. For the linear loss it decomposes across blocks, each optimized independently by picking $c_i = \arg\min_j -\mathbf{w}^t X_{ij\bullet}$ in overall $O(M\,C\,T)$ time. For general non-linear losses, we establish its computational hardness in Section 3.1. We

then develop three algorithms, Branch-and-Bound (Section 3.2), Block-Coordinate-Descent (Section 3.3), and a semidefinite relaxation (Section 3.4), together with a projected-gradient relaxation [7] baseline (Section 3.4.1) that our experiments found dominated by the other methods.

3.1 Computational Complexity

The optimization problem 1 has an exponential search space of size C^M. We characterize its complexity: the next two theorems show that it is NP-hard already in simple cases: the stabilizing loss at a single trait and the all-pass loss at two traits, each with two copies.

▶ **Theorem 3.** *The decision version of Problem 1 with the stabilizing-selection loss, namely deciding, given a tensor $\boldsymbol{X}$ and a threshold τ, whether there exists $\mathbf{c} \in [C]^M$ with $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) \le \tau$, is NP-complete, even for $T = 1$ and $C = 2$.*

Proof.

NP-membership. Given $\mathbf{c} \in [C]^M$, the score $\boldsymbol{X}_{\mathbf{c}} = \sum_{i=1}^{M} X_{ic_i\bullet}$ can be computed in $O(MT)$ time and the loss evaluated in $O(T)$ additional time, so the decision problem is in NP for any polynomial-time computable loss.

NP-hardness. We reduce from Partition[1]. Given positive integers $a_1, \ldots, a_M$ with $S = \sum_{i=1}^{M} a_i$, construct an instance with $C = 2$ copies, $T = 1$ trait, and the stabilizing loss $\mathcal{L}(x) = (x - S/2)^2$. Set the scores $X_{i11} = a_i$ and $X_{i21} = 0$ for all $i \in [M]$. For any selection vector $\mathbf{c} \in [2]^M$, the selected score is $\boldsymbol{X}_{\mathbf{c}} = \sum_{i:c_i=1} a_i$, and the loss is $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) = (\sum_{i:c_i=1} a_i - S/2)^2$. This equals zero if and only if $\sum_{i:c_i=1} a_i = S/2$, i.e. the integers $\{a_i : c_i = 1\}$ form an exact partition, so a poly-time solver for our decision problem would decide Partition, hence the problem is NP-hard and therefore NP-complete. ◀

▶ **Theorem 4.** *Problem 1 with the all-pass loss $\mathcal{L}(\mathbf{x}) = -P(\mathbf{D} = \mathbf{0}_T | \boldsymbol{X}_{\mathbf{c}} = \mathbf{x})$ is NP-hard for $T \ge 2$ and $C = 2$. When the residual covariance is diagonal, the objective factorizes into univariate Gaussian CDFs and is efficiently evaluable.*

Proof. We reduce from Partition. Given positive integers $a_1, \ldots, a_M$ with $S = \sum a_i$, set $T = 2$, $C = 2$, with scores $X_{i11} = a_i$, $X_{i21} = 0$, $X_{i12} = 0$, $X_{i22} = a_i$, symmetric thresholds $z_{K_1} = z_{K_2} = S/2$, and residual covariance $\sigma^2 \mathbf{I}_2$ (i.e. conditionally independent trait residuals with common variance σ^2), so the joint conditional probability factorises across traits. Under the liability-threshold model (eq. (13), Appendix B), each factor is the per-trait pass probability $P(D_k = 0 \mid \boldsymbol{X}_{\mathbf{c}}) = \Phi\big((\boldsymbol{X}_{\mathbf{c}_k} - z_{K_k})/\sigma\big)$, the complement of the crossing probability. For a selection with $A = \{i : c_i = 1\}$ and $u = \sum_{i \in A} a_i$ we have $\boldsymbol{X}_{\mathbf{c}_1} = u$ and $\boldsymbol{X}_{\mathbf{c}_2} = S - u$, so the loss is $-\Phi\big(\frac{u - S/2}{\sigma}\big) \cdot \Phi\big(\frac{S/2 - u}{\sigma}\big)$. Since $\Phi(z)\Phi(-z) = \Phi(z)(1 - \Phi(z))$ is uniquely maximized at $z = 0$, the loss achieves its minimum $-1/4$ if and only if $u = S/2$, i.e. an exact partition exists. ◀

We next show that the hardness above is only *weak*, because the problem is pseudo-polynomially solvable for any fixed number of traits.

[1] Partition: given positive integers $a_1, \ldots, a_n$ with $S = \sum a_i$, decide whether there exists $A \subseteq [n]$ with $\sum_{i \in A} a_i = S/2$. Partition is weakly NP-complete [17].

▶ **Proposition 5** (Pseudo-polynomial algorithm; weak hardness)**.** *Fix the number of traits T, assume integer scores X_{ijk}, and let $R_k \equiv \sum_{i=1}^{M}(\max_j X_{ijk} - \min_j X_{ijk})$ be the range of achievable sums for trait k, with $R = \max_k R_k$. Then Problem 1 with any loss $\mathcal{L} : \mathbb{Z}^T \to \mathbb{R}$ evaluable in $O(T)$ time is solved exactly in $O(M\,C\,R^T)$ time and $O(R^T)$ space. Hence for every fixed T the problem is pseudo-polynomial, so the NP-hardness of Theorems 3 and 4 is only* weak*, and bounded-precision score instances are solvable exactly in polynomial time.*

The proof, a dynamic program over the achievable score vectors, is given in Appendix A.2.

3.2 A Branch-and-Bound algorithm

In the Branch-and-Bound algorithm for selecting chromosomes *for monotone loss functions*, we grow a tree of all possible selected chromosomes, and at each level keep only leaves not dominated by other leaves. Finally, we evaluate the loss of all leaves at the last level. A tree of depth b is represented as a collection of paths from the root to the leaves $\Gamma = \{\mathbf{c}^{(1)}, .., \mathbf{c}^{(m)}\}$ where each $\mathbf{c}^{(j)} \in [C]^b$ represents the choices of chromosomes in the first b levels. The partial score sum is calculated: $\boldsymbol{X}_{\mathbf{c}^{(j)}} = \sum_{i=1}^{b} \boldsymbol{X}_{i\mathbf{c}_i^{(j)}\bullet}$, and dominated partial score vectors are pruned. Then, each of the remaining $\mathbf{c}^{(j)}$'s is expanded into C paths of length $b+1$. A formal step-by-step description is shown in Algorithm 1. The computational burden of a single run is determined by the number of non-dominated partial-sum vectors retained at each stage b, out of C^b possible partial selections. In the *worst case*, the Branch-and-Bound algorithm retains all of them, hence it may run in time exponential in M, as shown in the Appendix, Section A.3.

While the worst-case computational complexity of Algorithm 1 is exponential in M, the number of vectors considered may be far lower than C^M in practice. Further pruning can be achieved by bounding the loss with the coordinatewise *envelopes* of the score tensor: let $\boldsymbol{X}_\vee$ ($\boldsymbol{X}_\wedge$) denote the vector obtained by summing, over all chromosomes i, the coordinatewise maximum (minimum) of X_{ijk} across $j \in [C]$. Equivalently, writing $(\mathbf{A})_{\textcircled{\scriptsize$\vee$}}$ and $(\mathbf{A})_{\textcircled{\scriptsize$\wedge$}}$ for the column-wise maximum and minimum vectors of a matrix $\mathbf{A}$ (defined in Appendix A.1 and used in line 5 of Algorithm 1), we have $\boldsymbol{X}_\vee = \sum_i (\boldsymbol{X}_{i\bullet\bullet})_{\textcircled{\scriptsize$\vee$}}$ and $\boldsymbol{X}_\wedge = \sum_i (\boldsymbol{X}_{i\bullet\bullet})_{\textcircled{\scriptsize$\wedge$}}$. Since $\boldsymbol{X}_\wedge \preceq \boldsymbol{X}_{\mathbf{c}} \preceq \boldsymbol{X}_\vee$ for every feasible selection $\mathbf{c}$, and $\mathcal{L}$ is monotonically non-increasing, for the optimal selected score $\boldsymbol{X}_\star \equiv \boldsymbol{X}_{\mathbf{c}^\star}$ we obtain

$$\mathcal{L}(\boldsymbol{X}_\vee) \;\le\; \mathcal{L}(\boldsymbol{X}_\star) \;\le\; \mathcal{L}(\boldsymbol{X}_\wedge). \tag{3}$$

We call this an *envelope bound*: any partial-sum vector whose optimistic completion (the partial sum plus the remaining-block envelope $\sum_{l>i}(\boldsymbol{X}_{l\bullet\bullet})_{\textcircled{\scriptsize$\vee$}}$) yields a loss exceeding the feasible upper bound $\widehat{L}$ (the loss of the whole-genome optimum) cannot be optimal and is discarded. We refer to this discarding step as *envelope-pruning*.

For simulated problem instances, using the per-chromosome score model in eq. (11), the *average* number of leaves at each stage b was far below the C^b worst case over the regimes we tested (roughly $C^{b/2}$ for $T = 5$; see Figure 2(a,b)), which makes Algorithm 1 practical for problem instances of yeast and crop scale (e.g. $C = 2, M \le 21$).

3.2.1 Divide-and-conquer

We can improve the speed of our algorithm, by dividing the M chromosomes into groups, optimizing each of them separately, and then combining the solution in a manner where sub-vectors that cannot be extended to the optimal solution are filtered out. This procedure significantly improves performance while still being guaranteed to yield an optimal solution for monotone losses. Due to its technical details, it is described in Appendix A.4.

Algorithm 1 Choosing Best Genome (Branch-and-Bound).

Input: $\boldsymbol{X}$, a 3^{rd}-order tensor of polygenic scores.
Parameters: $\mathcal{L}$, a monotonically non-increasing loss function.

Γ ← {∅} with partial sum $\mathbf{0}_T$; $\widehat{L} \leftarrow \min_{j\in[C]} \mathcal{L}(\sum_i \boldsymbol{X}_{ij\bullet})$ (whole-genome optimum).
for $i = 1$ to M **do**
  EXPAND: $\Gamma \leftarrow \{(\mathbf{c}, j) : \mathbf{c} \in \Gamma,\ j \in [C]\}$ with partial sums $\boldsymbol{X}_{\mathbf{c}'} = \boldsymbol{X}_{\mathbf{c}} + X_{ij\bullet}$.
  PARETO-PRUNE: retain $\mathbf{c}' \in \Gamma$ only if $\nexists\, \tilde{\mathbf{c}} \in \Gamma$ with $\boldsymbol{X}_{\mathbf{c}'} \prec \boldsymbol{X}_{\tilde{\mathbf{c}}}$.
  (OPTIONAL) ENVELOPE-PRUNE: discard $\mathbf{c}' \in \Gamma$ if $\mathcal{L}\big(\boldsymbol{X}_{\mathbf{c}'} + \sum_{l>i} (\boldsymbol{X}_{l\bullet\bullet})_{\textcircled{\vee}}\big) > \widehat{L}$.
Output $\mathbf{c}^\star = \arg\min_{\mathbf{c}\in\Gamma} \mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$.

3.3 Block Coordinate Descent

Algorithm 1 (Branch-and-Bound) is inapplicable for non-monotone loss functions, and even for monotone losses its worst-case complexity is exponential in M. We therefore describe a simple *Block-Coordinate-Descent* (BCD) heuristic that applies to any loss and whose per-sweep complexity is linear in M, C, T.

The key observation is that updating a single block's selection c_i while holding all other $c_{i'}$ fixed changes only the contribution of block i to the aggregate score vector $\boldsymbol{X}_{\mathbf{c}} \in \mathbb{R}^T$. Concretely, write $\boldsymbol{X}_{\mathbf{c}} = \boldsymbol{X}_{\mathbf{c}}^{(-i)} + \boldsymbol{X}_{ic_i\bullet}$ where $\boldsymbol{X}_{\mathbf{c}}^{(-i)} \equiv \sum_{i'\neq i} \boldsymbol{X}_{i'c_{i'}\bullet}$ is the partial sum over all blocks except i. For each $j \in [C]$, evaluating $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}^{(-i)} + \boldsymbol{X}_{ij\bullet})$ costs only $O(T)$ once $\boldsymbol{X}_{\mathbf{c}}^{(-i)}$ is known. A full sweep over all blocks therefore costs $O(M\,C\,T)$ per iteration, and BCD typically converges to a 1-swap local optimum in $O(1)$ sweeps. In practice we repeat the procedure from R random restarts and keep the best result; see Algorithm 2.

Algorithm 2 Choosing Best Genome (Block Coordinate Descent).

Input: $\boldsymbol{X}$, a 3^{rd}-order tensor of polygenic scores.
Parameters: $\mathcal{L}$, any loss function; R, the number of restarts.

Build R initializations: i. whole-genome best $(j', \ldots, j')$ with $j' = \arg\min_j \mathcal{L}(\sum_i \boldsymbol{X}_{ij\bullet})$; ii. per-block greedy $(\arg\min_j \mathcal{L}(\boldsymbol{X}_{1j\bullet}), \ldots, \arg\min_j \mathcal{L}(\boldsymbol{X}_{Mj\bullet}))$; iii. $R-2$ uniformly random selections. Set $\mathbf{c}_{\text{best}} \leftarrow \emptyset$, $L_{\text{best}} \leftarrow +\infty$.
for each initialization $\mathbf{c}^{(0)}$ **do**
  $\mathbf{c} \leftarrow \mathbf{c}^{(0)}$; $\boldsymbol{X}_{\mathbf{c}} \leftarrow \sum_i \boldsymbol{X}_{ic_i\bullet}$; *changed* ← TRUE.
  while *changed* **do**
    changed ← FALSE.
    for $i = 1, \ldots, M$ **do**
      $\boldsymbol{X}_{\mathbf{c}}^{(-i)} \leftarrow \boldsymbol{X}_{\mathbf{c}} - \boldsymbol{X}_{ic_i\bullet}$; $j^\star \leftarrow \arg\min_{j\in[C]} \mathcal{L}(\boldsymbol{X}_{\mathbf{c}}^{(-i)} + \boldsymbol{X}_{ij\bullet})$.
      if $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}^{(-i)} + \boldsymbol{X}_{ij^\star\bullet}) < \mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$ **then**
        $c_i \leftarrow j^\star$; $\boldsymbol{X}_{\mathbf{c}} \leftarrow \boldsymbol{X}_{\mathbf{c}}^{(-i)} + \boldsymbol{X}_{ij^\star\bullet}$; *changed* ← TRUE.
  if $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) < L_{\text{best}}$ **then**
    $\mathbf{c}_{\text{best}} \leftarrow \mathbf{c}$; $L_{\text{best}} \leftarrow \mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$.
Output $\mathbf{c}_{\text{best}}$.

BCD has no global-optimality guarantee: it converges to a local optimum with respect to single-block swaps. However, the structural decoupling (independent blocks given the running score vector) makes it efficient enough that random restarts explore the landscape effectively in our yeast-scale simulations, where $R = 10$ restarts sufficed (Section 5).

3.4 A Semidefinite Relaxation

For the (non-monotone) stabilizing selection loss $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) = \sum_{k=1}^{T} w_k \boldsymbol{X}_{\mathbf{c}_k}^2$, the Branch-and-Bound algorithm of Section 3.2 does not apply and BCD has no optimality guarantee. We first review a simpler continuous relaxation that serves as a baseline, then develop a Shor-type semidefinite relaxation that provides a *certified lower bound* on the optimal loss, hence an *upper bound on the achievable gain*. Combined with a feasible heuristic such as BCD, the latter yields a certified optimality gap.

3.4.1 A simpler baseline: projected-gradient relaxation

As a standard baseline we also consider a continuous-relaxation approach: encode each selection $c_i \in [C]$ by a one-hot vector $\mathbf{C}_{i\bullet}$ and relax $\mathbf{C}_{i\bullet} \in \Delta_C$ (the C-simplex); the aggregate score becomes $\boldsymbol{X}_{\mathbf{c}} = [\boldsymbol{X} \times_2 \mathbf{C}]^t \mathbf{1}_M$. We minimize $\mathcal{L}\big([\boldsymbol{X} \times_2 \mathbf{C}]^t \mathbf{1}_M\big)$ over $\mathbf{C} \in \Delta_C^M$ by projected gradient descent with per-row simplex projection [7] and recover an integer solution by row-wise argmax; convergence guarantees apply only when $\mathcal{L}$ is convex [10]. For the stabilizing loss, a closed-form solution of the relaxation's Karush–Kuhn–Tucker (KKT) optimality conditions gives a Lagrangian variant. Empirically the rounded loss was worse than the trivial baseline of picking copy 1 for every block (Section 5): the relaxed optimum sits in the simplex interior, and argmax rounding then picks essentially-arbitrary copies. This failure mode directly motivates the semidefinite relaxation below, which enforces an integer-feasible structure and yields a certified lower bound on the integer loss.

3.4.2 The semidefinite relaxation

Let $\mathbf{q} = \mathrm{vec}(\mathbf{C}) \in \{0,1\}^{MC}$ be the vectorized one-hot selection matrix ($C_{ij} = 1$ iff $c_i = j$; vec as defined in Appendix A.1), so $q_{(i-1)C+j} = C_{ij}$. The stabilizing loss is then the quadratic form $\mathcal{L}(\mathbf{q}) = \mathbf{q}^t \mathbf{A} \mathbf{q}$, where $\mathbf{A} = \sum_{k=1}^{T} w_k \mathbf{x}_k \mathbf{x}_k^t$ and $\mathbf{x}_k = \mathrm{vec}(\boldsymbol{X}_{\bullet\bullet k})$. Following Shor [39], we lift the binary problem with the bordered matrix $\boldsymbol{W} = \begin{pmatrix} 1 & \mathbf{q}^t \\ \mathbf{q} & \mathbf{Q} \end{pmatrix} \in \mathbb{R}_{(MC+1)\times(MC+1)}$. A genuine selection corresponds to a *rank*-1 $\boldsymbol{W} \succeq 0$ with $\mathbf{Q} = \mathbf{q}\mathbf{q}^t$, so that $\mathcal{L}(\mathbf{q}) = \mathrm{tr}(\mathbf{A}\mathbf{Q})$; imposing this rank-1 condition, the binary constraint $\mathrm{diag}(\mathbf{Q}) = \mathbf{q}$ (valid since $q_i^2 = q_i$ for binary q_i), and the per-block row-sum constraints reformulates Problem 1 exactly. Dropping the (non-convex) rank-1 constraint yields the SDP relaxation

$$\begin{aligned} \min_{\boldsymbol{W}} \quad & \mathrm{tr}(\mathbf{A}\mathbf{Q}) \\ \text{subject to} \quad & \boldsymbol{W} \succeq 0, \quad W_{1,1} = 1, \quad \mathrm{diag}(\mathbf{Q}) = \mathbf{q}, \\ & \sum_{j=1}^{C} q_{(i-1)C+j} = 1 \text{ for each } i \in [M]. \end{aligned} \tag{4}$$

A feasible $\boldsymbol{W}$ need not be rank 1 (so the optimum may be attained at a $\boldsymbol{W}$ that is not a genuine selection); but every rank-1 integer-feasible $\boldsymbol{W}$ remains feasible, so the optimal value $L^\star_{\mathrm{SDP}}$ is a *lower bound* on the integer minimum loss, $L^\star_{\mathrm{SDP}} \le \min_{\mathbf{c}} \mathcal{L}(\boldsymbol{X}_{\mathbf{c}})$. The relaxation is solved in polynomial time using interior-point methods (we use CSDP [5]). Given any feasible integer solution $\hat{\mathbf{c}}$ (e.g. from BCD), the certified optimality gap is $\mathcal{L}(\boldsymbol{X}_{\hat{\mathbf{c}}}) - L^\star_{\mathrm{SDP}}$. The procedure is summarised in Algorithm 3.

The SDP is used solely to certify the lower bound; a feasible integer selection could be recovered by block-wise argmax rounding of $\mathbf{q}^\star$, but empirically BCD with random restarts (Algorithm 2) produced a stronger feasible solution at much lower computational cost in our experiments (Section 5), so we pair the SDP bound with the BCD solution.

Algorithm 3 Certified Lower Bound via Semidefinite Relaxation.

Input: $\boldsymbol{X}$, a 3^{rd}-order tensor of polygenic scores; weights $\mathbf{w}$.
Output: $L^{\star}_{\mathrm{SDP}}$, a lower bound on the optimal stabilizing loss.

1: $\mathbf{A} \leftarrow \sum_{k=1}^{T} w_k \,\mathrm{vec}(\boldsymbol{X}_{\bullet\bullet k})\,\mathrm{vec}(\boldsymbol{X}_{\bullet\bullet k})^t$.
2: Solve the SDP in eq. (4) for an optimal $\boldsymbol{W}^{\star} = \begin{pmatrix} 1 & (\mathbf{q}^{\star})^t \\ \mathbf{q}^{\star} & \mathbf{Q}^{\star} \end{pmatrix}$, using an interior-point solver such as CSDP.
3: Output the bound $L^{\star}_{\mathrm{SDP}} = \mathrm{tr}(\mathbf{A}\mathbf{Q}^{\star})$.

4 Expected Gain: Analytic Prediction

To anchor the simulations that follow, we first derive a closed-form prediction for the expected gain under a population-level statistical model. Suppose the score tensor $\boldsymbol{X}$ is drawn from the matrix-Normal model of Appendix B, in which the score vector for each chromosome i has covariance $\alpha_i^2\,\boldsymbol{\Sigma}^{(\boldsymbol{X})}$ with $\sum_i \alpha_i^2 = 1$, and blocks are independent. For the linear loss $\mathcal{L} = -\mathbf{w}^t \boldsymbol{X}_{\mathbf{c}}$ with $\mathbf{w} \succeq \mathbf{0}_T$ and i.i.d. source genomes ($\boldsymbol{\Sigma}^{(\boldsymbol{C})} = \mathbf{I}_C$), standard extreme-value arguments [22] give whole-genome and chromosomal expected gains

$$\begin{aligned} \mathbb{E}_{\boldsymbol{X}} G_e &\approx 0.77\sqrt{\mathbf{w}^t \boldsymbol{\Sigma}^{(\boldsymbol{X})} \mathbf{w}}\sqrt{\log C}, \\ \mathbb{E}_{\boldsymbol{X}} G &\approx \Big(\sum\nolimits_{i=1}^{M} \alpha_i\Big)\, 0.77\sqrt{\mathbf{w}^t \boldsymbol{\Sigma}^{(\boldsymbol{X})} \mathbf{w}}\sqrt{\log C}, \end{aligned} \tag{5}$$

so the ratio of expected gains is $\sum_i \alpha_i$. Using chromosome lengths as a proxy for α_i^2 (Table 2, Appendix B), this gives a predicted $\sum_i \alpha_i \approx 3.89$-fold improvement for yeast ($M = 16$) and ≈ 4.68-fold for human ($M = 23$, 22 autosomes plus the X chromosome). For non-linear losses the closed form no longer applies and we evaluate the gain numerically below.

5 Simulation Results

To examine the utility of the algorithms, we have implemented them in an R package available at *github* (see Appendix C). We simulated chromosomal scores from the statistical model of Appendix B, a Matrix-Gaussian distribution [19], with parameters inspired by a yeast-like scenario: $M = 16$ chromosomes (relative lengths in Table 2), varying C, and $T = 5$ quantitative traits under a threshold model. Each trait has a "failure" probability of 0.1 (analogous to a disease prevalence or a probability of falling below a quality threshold), a narrow-sense heritability $h^2 = 50\%$, and polygenic scores explaining 20% of the liability variance; the residual (non-score) covariance $\boldsymbol{\Sigma}^{(\boldsymbol{Y})} + \boldsymbol{\Sigma}^{(\boldsymbol{e})}$ therefore has diagonal 0.8, with off-diagonal entries set to 0.65 (moderate cross-trait residual correlation).

We used the sum of threshold-trait probability loss function with equal weights, and the (minus) probability of passing all thresholds simultaneously. The baseline loss under random selection was, as expected, $0.1 \times 5 = 0.5$ for the first loss. For the all-pass criterion, the baseline pass probability $P(\mathbf{D} = \mathbf{0}_T)$ was 0.72 (so the corresponding loss is -0.72), above $0.9^5 \approx 0.59$ that would hold under trait independence.

Tree-pruning behavior of Branch-and-Bound (Figure 2a,b). For a single simulated instance with $M = 16$, $C = 2$, $T = 5$, we tracked the number of Pareto-optimal partial-sum vectors retained at each stage of Algorithm 1 (Branch-and-Bound) and Algorithm 4 (divide-and-conquer). Branch-and-Bound retained $\approx$ 3,000 vectors at the last stage, and the divide-and-conquer variant retained only ≈ 100, roughly two orders of magnitude fewer than the exhaustive $2^{16} = 65{,}536$ leaves. Averaging over 100 simulations for varying $M \in [2, 16]$,

the final Pareto-set size for divide-and-conquer remained in the few-hundreds range even at $M = 16$. The favorable average-case behavior can be understood through the theory of random Pareto fronts: when C^b i.i.d. vectors in $\mathbb{R}^T$ are drawn, the expected number of Pareto-optimal points is $O\big((\log C^b)^{T-1}\big) = O\big(b^{T-1}(\log C)^{T-1}\big)$ [2], which for fixed T grows polynomially in b. The partial-sum vectors in Algorithm 1 are not i.i.d., but their approximate independence under the per-chromosome Matrix-Normal distribution in eq. (11) suggests a similar scaling, consistent with the empirically observed growth of $\approx C^{b/2}$ non-dominated vectors for $T = 5$. For $T \gg 5$ the Pareto front grows and the tree may become prohibitively large; in that regime BCD or the SDP-bound become attractive.

Threshold loss (Figure 2c). For the weighted-sum threshold loss with $M = 16$, $T = 5$ and $C \in [2, 10]$, we ran 12 independent simulations and compared the certified Branch-and-Bound optimum[2] (Algorithm 4) against (i.) Block-Coordinate-Descent with $R = 10$ restarts (Algorithm 2) and (ii.) whole-genome selection ($\min_{j\in[C]} \mathcal{L}(\sum_i \boldsymbol{X}_{ij\bullet})$). BCD matched the Branch-and-Bound optimum on all 108 simulated instances (12 replicates per C, $C \in [2, 10]$); mean BCD running time was 0.01 s vs. 4.8 s for Branch-and-Bound at $C = 10$ (all timings in this section use the divide-and-conquer variant, Algorithm 4); summed over all $C \in [2, 10]$, BCD's total running time was 466× shorter (the per-C speed-up grows with C as Branch-and-Bound slows). To confirm these conclusions are not specific to $M = 16, T = 5$, we repeated this comparison across $M \in \{8, 16, 23\}$ and $T \in \{2, 3, 5\}$ ($C = 5$, 10 replicates each; 90 instances); BCD with $R = 10$ restarts again matched the certified Branch-and-Bound optimum on every instance. Branch-and-Bound time grew with problem size, reaching ≈1.7 s at $M = 23$ and over an hour per instance at $T = 6$ (as the Pareto front expands), while BCD stayed ≈0.01 s throughout. Chromosomal selection achieved roughly 3× the expected gain of whole-genome selection at every value of C tested. This empirical ratio is consistent with the analytic prediction of $\sum_i \alpha_i \approx 3.89$ for yeast derived in eq. (5) for the linear loss: although the threshold loss is non-linear, over the simulated range it behaves approximately linearly in the relevant score directions, giving a similar block-additive geometry, so the ratio of expected gains tracks the linear prediction closely. The full agreement between BCD and B&B suggests that instances drawn from the matrix-Normal model in eq. (11) induce a benign optimization landscape for BCD, where $R = 10$ restarts suffice with high probability. On adversarial inputs (e.g. PARTITION-style constructions, cf. Theorem 3) BCD's 1-swap neighbourhood is no longer sufficient and the worst-case gap can be large. The projected-gradient baseline (Section 3.4.1) was again worse than the trivial copy-1 selection and is not displayed in Figure 2c.

Stabilizing loss (Figure 2d). For the (non-monotone) stabilizing loss with the same yeast parameters and 30 simulations per C, we computed (i) the BCD solution (Algorithm 2) and (ii) the SDP-guaranteed lower bound (Algorithm 3). Across all 270 simulated instances (30 replicates per C, $C \in [2, 10]$) the SDP bound was below BCD's feasible loss. Averaging across replicates, BCD's certified optimality gap, the relative excess of its feasible loss over the SDP lower bound $(\mathcal{L}_{\mathrm{BCD}} - L^\star_{\mathrm{SDP}})/L^\star_{\mathrm{SDP}}$, was 2.5%, 4.1%, 10.1% and 8.6% at $C = 2, 3, 4, 5$, respectively; at $C \geq 6$ the SDP bound and the BCD loss both approach zero and the absolute gap shrinks below 10^{-3}. Note that Figure 2(d) plots the resulting expected *gain*, on whose scale this gap is under 1%, so the BCD and SDP-ceiling curves are visually indistinguishable.

[2] Our Branch-and-Bound implementation supports an optional heuristic cap of B retained Pareto-optimal vectors per stage (Appendix C); for all experiments reported in this section the cap was set to $B = 10^7$ and was never reached, hence the reported solutions are genuine global optima of Problem 1.

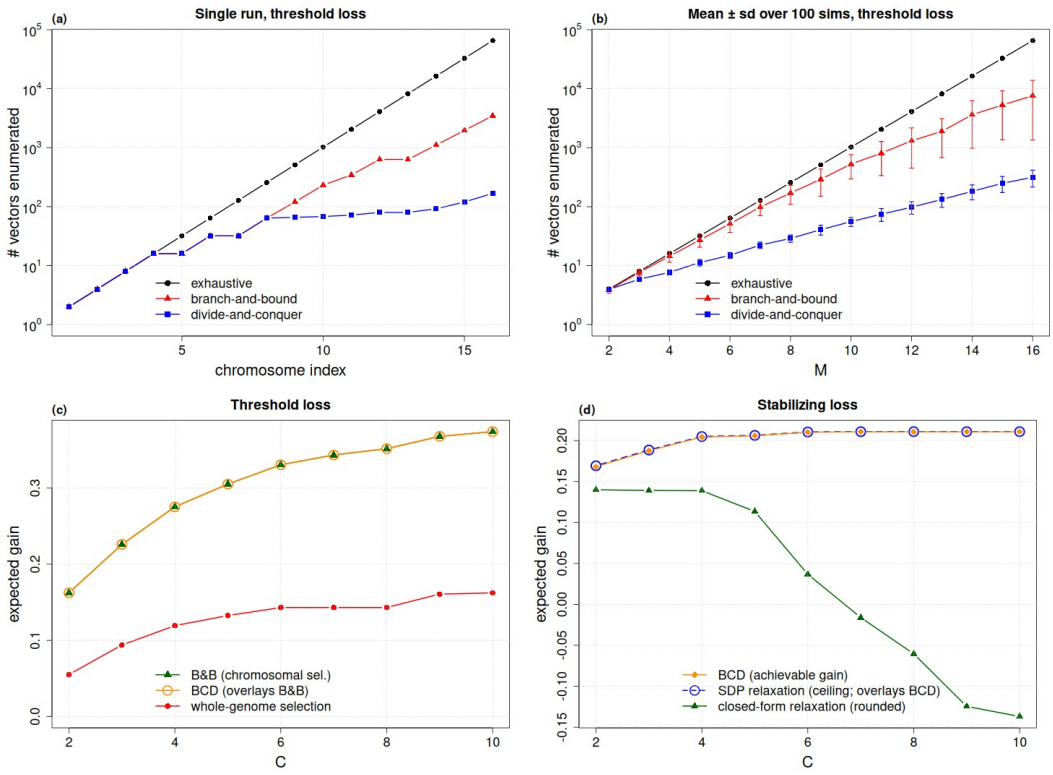

Figure 2 (a.) Number of Pareto-optimal partial-sum vectors retained at each stage (y-axis) vs. the chromosome index (x-axis), for a single run with $M = 16$ chromosomes (yeast), $C = 2$ source strains, and $T = 5$ traits, using the threshold loss. Black: exhaustive search (C^b); red: Algorithm 1 (Branch-and-Bound); blue: the divide-and-conquer variant (Algorithm 4), here using $b = 2$ sub-blocks of $\lfloor M/2 \rfloor$ chromosomes each (two halves of 8 at $M = 16$). Divide-and-conquer retains ≈ 100 vectors at $M = 16$, roughly two orders of magnitude below exhaustive. (b.) Mean $\pm$ one standard deviation of the final Pareto-set size across 100 simulations, for varying $M \in [2, 16]$. Both Branch-and-Bound (red) and divide-and-conquer (blue) grow much more slowly than exhaustive search over the tested regime; divide-and-conquer remains in the few-hundreds range at $M = 16$. (c.) Expected gain on the threshold loss vs. the number of source strains C, for yeast parameters ($M = 16$, $T = 5$, 12 simulations per C). Green: Branch-and-Bound (certified global optimum); orange: BCD (Algorithm 2, $R = 10$ restarts), which matched the Branch-and-Bound optimum on every simulated instance and is visually indistinguishable; red: whole-genome selection. Chromosomal selection (B&B or BCD) achieves $\approx 3\times$ the expected gain of whole-genome selection. (d.) Expected gain on the (non-monotone) stabilizing loss vs. C, for the same parameters and 30 simulations per C. Orange: BCD (achievable gain); blue dashed (x markers): SDP relaxation (Algorithm 3) value, an upper bound on the achievable gain; green: closed-form Lagrangian relaxation followed by argmax rounding. BCD's certified optimality gap relative to the SDP loss bound is 2–10% for $C \leq 5$; on the gain scale plotted here that gap is under 1%, so the BCD (orange) and SDP (blue) curves nearly coincide. The closed-form relaxation degrades sharply and yields negative gain at $C \geq 6$.

In contrast, the closed-form Lagrangian relaxation (dropping the nonneg constraint, solving the row-sum-constrained quadratic in closed form, then rounding by row-wise argmax; plotted in green) degrades sharply with C and yields negative gain (worse than the trivial baseline) at $C \geq 6$, illustrating the failure mode of simplex-relaxation+rounding for this loss.

Practical recommendation. Based on these results, we recommend BCD with random restarts as the default algorithm in practice. Branch-and-Bound (Section 3.2) should be used when a certified global optimum is required for a monotone loss, and is tractable at the yeast/crop scale ($M \lesssim 21$, $C \lesssim 10$); at larger scale (e.g. when blocks are taken to be finer-grained chromosomal segments, $M \gg 21$, or when the Pareto front blows up at large T), the certified branch is infeasible and BCD becomes the only practical option. The SDP lower bound (Section 3.4) provides an optimality-gap certificate for quadratic losses; its complexity grows as $(MC)^{O(1)}$ via interior-point methods, so it remains usable beyond the regime where B&B applies, though it too eventually scales out and one must fall back on BCD alone.

6 Discussion

We have defined and formulated the genomic block selection problem and provided three algorithms with complementary roles. Our Branch-and-Bound algorithm (Algorithm 1) returns the certified global optimum for any monotone loss; despite worst-case exponential complexity, the divide-and-conquer variant (Algorithm 4) kept the retained Pareto set in the few-hundreds range at $M = 16$, while the plain version retained $\approx$ 3,000 vectors, both well within reach for yeast and crop scenarios ($M \leq 21$). The Block-Coordinate-Descent heuristic (Algorithm 2) applies to any loss, runs in milliseconds on yeast-scale instances, and was empirically optimal on every threshold-loss instance we tested (all 108 simulated instances). For the non-monotone stabilizing loss the SDP relaxation (Algorithm 3) provides a certified lower bound that, combined with the BCD solution, yielded a certified optimality gap of $\lesssim 10\%$ for $C \leq 5$ in our simulations. The standard projected-gradient relaxation (Section 3.4.1), included as a baseline, was strictly dominated by BCD on both monotone and non-monotone losses in our setup. Establishing problem-instance-dependent approximation guarantees for BCD and tightening the SDP at larger C are interesting directions for future research.

The problem is motivated by several current and emerging technologies. In crop breeding, chromosome substitution lines from wild relatives are routinely used to improve polygenic traits in wheat and other polyploid crops [16, 33], yet the optimal combination of donor segments for multiple traits is typically chosen by breeder intuition rather than formal optimization; our algorithms apply directly once polygenic scores are available. In yeast, the Sc2.0 synthetic genome project [34] and panels of sequenced strains [31] make chimeric genomes for industrial phenotypes feasible (across dozens of yeast growth traits, including high-temperature growth, mapped QTL explain a median of $\approx 73\%$ of the additive heritability [3]). In mammalian cells, high-fidelity chromosome transplantation has been demonstrated [32], suggesting future relevance to animal models and potentially human applications, subject to technological and ethical constraints.

Our framework assumes additive polygenic scores across chromosomes, a consequence of the infinitesimal model; cross-chromosome epistasis, particularly between divergent source genomes, could reduce the realized gain, so eq. (5) should be read as an optimistic benchmark under the additive model rather than a worst-case bound. Our empirical evaluation, moreover, uses scores simulated from the matrix-Normal model (calibrated to real yeast chromosome

lengths and heritabilities); validating the algorithms on empirical per-chromosome polygenic scores (e.g. from sequenced yeast strain panels or crop populations) is an important next step. Extending the framework to higher-order chromosomal interactions, finer-grained selection units (chromosomal segments rather than whole chromosomes), and nonlinear polygenic scores are all important directions for future work.

References

1 B.W. Bader and T.G. Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software (TOMS)*, 32(4):635–653, 2006. doi:10.1145/1186785.1186794.

2 O. Barndorff-Nielsen and M. Sobel. On the distribution of the number of admissible points in a vector random sample. *Theory of Probability & Its Applications*, 11(2):249–269, 1966. doi:10.1137/1111020.

3 J.S. Bloom, J. Boocock, S. Treusch, M.J. Sadhu, L. Day, H. Oates-Barker, and L. Kruglyak. Rare variants contribute disproportionately to quantitative trait variation in yeast. *eLife*, 8:e49212, 2019. doi:10.7554/eLife.49212.

4 J.S. Bloom, I.M. Ehrenreich, W.T. Loo, T.-L.V. Lite, and L. Kruglyak. Finding the sources of missing heritability in a yeast cross. *Nature*, 494(7436):234–237, 2013.

5 B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1–4):613–623, 1999.

6 B. Bulik-Sullivan, H.K. Finucane, V. Anttila, A. Gusev, F.R. Day, P-R. Loh, L. Duncan, J.RB. Perry, N. Patterson, E.B. Robinson, et al. An atlas of genetic correlations across human diseases and traits. *Nature genetics*, 47(11):1236–1241, 2015.

7 Y. Chen and X. Ye. Projection onto a simplex. *arXiv preprint*, 2011. arXiv:1101.6081.

8 J.M. Cherry, E.L. Hong, C. Amundsen, R. Balakrishnan, G. Binkley, E.T. Chan, et al. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Research*, 40(D1):D700–D705, 2012.

9 S.W. Choi, T. S-H. Mak, and P.F. O'Reilly. Tutorial: a guide to performing polygenic risk score analyses. *Nature Protocols*, 15(9):2759–2772, 2020.

10 R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(1):261–275, 1993. doi:10.1007/BF01585170.

11 J. Crossa, P. Pérez-Rodríguez, J. Cuevas, O. Montesinos-López, D. Jarquín, G. de los Campos, J. Burgueño, J.M. González-Camacho, S. Pérez-Elizalde, Y. Beyene, et al. Genomic selection in plant breeding: methods, models, and perspectives. *Trends in Plant Science*, 22(11):961–975, 2017.

12 M.G. Durrant, N.T. Perry, J.J. Pai, A.R. Jangid, J.S. Athukoralage, M. Hiraizumi, J.P. McSpedon, A. Pawluk, H. Nishimasu, S. Konermann, and P.D. Hsu. Bridge RNAs direct programmable recombination of target and donor DNA. *Nature*, 630:984–993, 2024. doi:10.1038/s41586-024-07552-4.

13 D. Eddelbuettel and R. François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. doi:10.18637/jss.v040.i08.

14 D.S. Falconer. *Introduction to Quantitative Genetics*. Pearson Education India, 1996.

15 H.K. Finucane, B. Bulik-Sullivan, A. Gusev, G. Trynka, Y. Reshef, P-R. Loh, V. Anttila, H. Xu, C. Zang, K. Farh, et al. Partitioning heritability by functional annotation using genome-wide association summary statistics. *Nature Genetics*, 47(11):1228, 2015.

16 B. Friebe, J. Jiang, W.J. Raupp, R.A. McIntosh, and B.S. Gill. Characterization of wheat-alien translocations conferring resistance to diseases and pests: current status. *Euphytica*, 91(1):59–87, 1996.

17 M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

18 T. Ge, C-Y. Chen, Y. Ni, Y-C. Anne Feng, and J.W. Smoller. Polygenic prediction via Bayesian regression and continuous shrinkage priors. *Nature Communications*, 10(1):1–10, 2019.

19 A.K. Gupta and D.K. Nagar. *Matrix Variate Distributions*. Chapman and Hall/CRC, 2018.

20 T.F. Hansen. Stabilizing selection and the comparative analysis of adaptation. *Evolution*, 51(5):1341–1351, 1997.

21 M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J.A. Doudna, and E. Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *Science*, 337(6096):816–821, 2012.

22 E. Karavani, O. Zuk, D. Zeevi, G. Atzmon, N. Barzilai, N.C. Stefanis, A. Hatzimanolis, N. Smyrnis, D. Avramopoulos, L. Kruglyak, et al. Screening human embryos for polygenic traits has limited utility. *Cell*, 179(6):1424–1435, 2019.

23 A.V. Khera, M. Chaffin, K.G. Aragam, M.E. Haas, C. Roselli, S.H. Choi, P. Natarajan, E.S. Lander, S.A. Lubitz, P.T. Ellinor, et al. Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. *Nature Genetics*, 50(9):1219–1224, 2018.

24 T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi:10.1137/07070111X.

25 S.A Lambert, L. Gil, S. Jupp, S.C. Ritchie, Y. Xu, A. Buniello, A. McMahon, G. Abraham, M. Chapman, H. Parkinson, et al. The polygenic score catalog as an open database for reproducibility and systematic evaluation. *Nature Genetics*, 53(4):420–425, 2021.

26 T. Lencz, D. Backenroth, E. Granot-Hershkovitz, A. Green, K. Gettler, J.H. Cho, O. Weissbrod, O. Zuk, and S. Carmi. Utility of polygenic embryo screening for disease depends on the selection strategy. *Elife*, 10:e64716, 2021.

27 T.H.E. Meuwissen, B.J. Hayes, and M.E. Goddard. Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157(4):1819–1829, 2001.

28 J. Pattee and W. Pan. Penalized regression and model selection methods for polygenic scores on summary statistics. *PLoS Computational Biology*, 16(10):e1008271, 2020. doi:10.1371/JOURNAL.PCBI.1008271.

29 M. Paulis, A. Castelli, L. Susani, M. Lizier, I. Lagutina, M.L. Focarelli, C. Recordati, P. Uva, F. Faggioli, T. Neri, et al. Chromosome transplantation as a novel approach for correcting complex genomic disorders. *Oncotarget*, 6(34):35218–35230, 2015.

30 M. Paulis, L. Susani, A. Castelli, T. Suzuki, T. Hara, L. Straniero, S. Duga, D. Strina, S. Mantero, E. Caldana, et al. Chromosome transplantation: A possible approach to treat human x-linked disorders. *Molecular Therapy-Methods & Clinical Development*, 17:369–377, 2020.

31 J. Peter, M. De Chiara, A. Friedrich, J.-X. Yue, D. Pflieger, A. Bergström, A. Sigwalt, B. Barre, K. Freel, A. Llored, et al. Genome evolution across 1,011 *Saccharomyces cerevisiae* isolates. *Nature*, 556(7701):339–344, 2018.

32 G. Petris, S. Grazioli, L. van Bijsterveldt, P. Murat, K.C. Liu, J. Birnbaum, J.E. Sale, and J.W. Chin. High-fidelity human chromosome transfer and elimination. *Science*, 390:1038–1043, 2025. doi:10.1126/science.adv9797.

33 L.L. Qi, B. Friebe, P. Zhang, and B.S. Gill. Homoeologous recombination, chromosome engineering and crop improvement. *Chromosome Research*, 15(1):3–19, 2007.

34 S.M. Richardson, L.A. Mitchell, G. Stracquadanio, K. Yang, J.S. Dymond, J.E. DiCarlo, D. Lee, C.L.V. Huang, S. Chandrasegaran, Y. Cai, J.D. Boeke, and J.S. Bader. Design of a synthetic yeast genome. *Science*, 355(6329):1040–1044, 2017.

35 J.S. Sanjak, J. Sidorenko, M.R. Robinson, K.R. Thornton, and P.M. Visscher. Evidence of directional and stabilizing selection in contemporary humans. *Proceedings of the National Academy of Sciences*, 115(1):151–156, 2018.

36 I.I. Schmalhausen. *Factors of evolution: the theory of stabilizing selection.* Blakiston, 1949.

37 V.A. Schneider, T. Graves-Lindsay, K. Howe, N. Bouk, H.-C. Chen, P.A. Kitts, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Research*, 27(5):849–864, 2017.

38 Y. Shen, G. Stracquadanio, Y. Wang, K. Yang, L.A. Mitchell, Y. Xue, Y. Cai, T. Chen, J.S. Dymond, K. Kang, et al. SCRaMbLE generates designed combinatorial stochastic diversity in synthetic chromosomes. *Genome Research*, 26(1):36–49, 2016.

39 N.Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25(6):1–11, 1987.

40 P. M. Visscher, N. R. Wray, Q. Zhang, P. Sklar, M. I. McCarthy, M. A. Brown, and J. Yang. 10 years of GWAS discovery: Biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017.

41 J. Yang, T.A. Manolio, L.R. Pasquale, E. Boerwinkle, N. Caporaso, J.M. Cunningham, M. De Andrade, B. Feenstra, E. Feingold, M.G. Hayes, et al. Genome partitioning of genetic variation for complex traits using common snps. *Nature Genetics*, 43(6):519, 2011.

A Algorithms and Optimization Details

A.1 Notations

We use a single *block-by-block* vectorization throughout the paper. For a matrix $X \in \mathbb{R}_{m\times n}$, $\mathrm{vec}(X) \in \mathbb{R}^{mn}$ is defined by $[\mathrm{vec}(X)]_{(i-1)n+j} = X_{ij}$ (stacking the rows $1,\ldots,m$ in order). Similarly, for a 3^{rd}-order tensor $X \in \mathbb{R}_{m\times n\times p}$, $\mathrm{mat}(X) \in \mathbb{R}_{mn\times p}$ has row $(i-1)n+j$ equal to the fiber $X_{ij\bullet}$.

There are 2^T possible binary vectors of length T, with each such vector $\mathbf{d} \in \{0,1\}^T$, corresponding to an orthant $O_{\mathbf{d}} \subset \mathbb{R}^T$ defined as $O_{\mathbf{d}} \equiv \Big\{(x_1,..,x_T)\text{s.t.}(-1)^{d_i}x_i < 0, \forall i \in [T]\Big\}$. These 2^T orthants form a disjoint union of $\mathbb{R}^T$ (ignoring equalities with the axes).

In similar to eq. (1), for a vector $\mathbf{V} \in \mathbb{R}^p$, define the *3-mode* tensor-by-vector product as the matrix $[\boldsymbol{X} \times_3 \mathbf{V}] \in \mathbb{R}_{m\times n}$, with elements defined by:

$$[\boldsymbol{X} \times_3 \mathbf{V}]_{ij} \equiv \sum_{k=1}^{p} X_{ijk}V_k,\ \forall i \in [m], \forall j \in [n]. \tag{6}$$

Element-wise notations. For two vectors or matrices A, B of the same size, their Hadamard product $A \odot B$ is the entrywise product (e.g. $[A \odot B]_{ij} = a_{ij}b_{ij}$ for matrices). For a matrix A, the column-wise maximum and minimum vectors are denoted $A_{\bigvee}, A_{\bigwedge}$, with $[A_{\bigvee}]_j = \max_i a_{ij}$ and $[A_{\bigwedge}]_j = \min_i a_{ij}$ (the max/min over rows, for each column).

A.2 Proof of the pseudo-polynomial algorithm

Proof of Proposition 5. Shift each trait so that $\min_j X_{ijk} = 0$ for every i, k; this translates every achievable score vector, and the loss argument, by a fixed vector, leaving the minimizer unchanged. Every partial score vector then lies in the grid $\prod_{k=1}^{T}\{0,\ldots,R_k\}$. Let S_i be the set of partial-sum vectors attainable using blocks $1,\ldots,i$: $S_0 = \{\mathbf{0}_T\}$ and $S_i = \{\boldsymbol{v} + X_{ij\bullet} : \boldsymbol{v} \in S_{i-1},\ j \in [C]\}$. Each S_i is a subset of the grid, so $|S_i| \le \prod_k (R_k+1)$, and computing $S_1,\ldots,S_M$ costs $O(M\,C\,\prod_k R_k)$; returning $\min\{\mathcal{L}(\boldsymbol{v}) : \boldsymbol{v} \in S_M\}$ gives the exact optimum. As $\prod_k R_k$ is polynomial in the largest score magnitude for fixed T, this is a pseudo-polynomial algorithm. ◀

A.3 Worst-case behavior of Branch-and-Bound

▶ **Proposition 6.** *In the worst case, the number of non-dominated vectors at stage b of Algorithm 1 is C^b.*

Proof. We construct an instance with $T = 2$ traits as follows. Draw $u_{ij} \overset{i.i.d.}{\sim} U[0,1]$ for $i \in [M], j \in [C]$, and set $X_{ij\bullet} \equiv (u_{ij}, -u_{ij})$. For any selection $\mathbf{c} \in [C]^b$, the partial sum is $\sum_{i=1}^{b} X_{ic_i\bullet} = (s(\mathbf{c}), -s(\mathbf{c}))$, where $s(\mathbf{c}) \equiv \sum_{i=1}^{b} u_{ic_i}$. With probability 1 over the draws, the C^b values $\{s(\mathbf{c}) : \mathbf{c} \in [C]^b\}$ are all distinct. For two selections $\mathbf{c} \neq \mathbf{c}'$ with $s(\mathbf{c}) < s(\mathbf{c}')$ the partial sums $(s(\mathbf{c}), -s(\mathbf{c}))$ and $(s(\mathbf{c}'), -s(\mathbf{c}'))$ are coordinatewise incomparable: the first has a smaller first coordinate but a larger second coordinate. Hence all C^b partial sums are pairwise Pareto-incomparable and Algorithm 1 retains all of them at stage b. ◀

A.4 The divide-and-conquer algorithm

Partition the M chromosomes into b disjoint sub-blocks $B_1, \ldots, B_b$ with $\uplus_i B_i = [M]$ and run Algorithm 1 on each sub-tensor $\boldsymbol{X}_{B_i\bullet\bullet}$ to obtain the Pareto-optimal set $\Gamma^{(i)}$ and sub-block optimum $\boldsymbol{X}^{(i)}_{\star}$. Writing $\boldsymbol{X}^{(i)}_{\textcircled{\vee}}$ ($\boldsymbol{X}^{(i)}_{\textcircled{\wedge}}$) for the coordinatewise max (min) over $\Gamma^{(i)}$, the global optimum satisfies $\mathcal{L}(\sum_i \boldsymbol{X}^{(i)}_{\textcircled{\vee}}) \le \mathcal{L}(\boldsymbol{X}_\star) \le \mathcal{L}(\sum_i \boldsymbol{X}^{(i)}_{\star}) \le \mathcal{L}(\sum_i \boldsymbol{X}^{(i)}_{\textcircled{\wedge}})$ for any monotonically non-increasing $\mathcal{L}$. The first inequality gives a lower-bound certificate; the second gives the best loss found so far, $\widehat{L} = \mathcal{L}(\sum_i \boldsymbol{X}^{(i)}_{\star})$, used for envelope pruning (line 7 of Algorithm 4). Algorithm 4 combines per-sub-block enumeration with Pareto- and envelope-pruning during the merge phase.

Algorithm 4 Choosing Best Genome (Branch-and-Bound Divide-and-Conquer).

Input: $\boldsymbol{X}$, a 3^{rd}-order tensor of polygenic scores.
Parameters: $\mathcal{L}$, a monotonically non-increasing loss; $\{B_1, \ldots, B_b\}$, a partition of $[M]$.
1: **for** $i \in [b]$ **do**
2: Run Algorithm 1 on $\boldsymbol{X}_{B_i\bullet\bullet}$ to get Pareto set $\Gamma^{(i)} = \{\boldsymbol{X}^{(i,k)} : k \in [S_i]\}$, sub-block optimum $\boldsymbol{X}^{(i)}_{\star}$, and envelopes $\boldsymbol{X}^{(i)}_{\textcircled{\vee}}, \boldsymbol{X}^{(i)}_{\textcircled{\wedge}}$.
3: $\widehat{L} \leftarrow \mathcal{L}\big(\sum_{i=1}^{b} \boldsymbol{X}^{(i)}_{\star}\big)$; $\mathcal{M}^{(0)} \leftarrow \{\emptyset\}$ with partial sum $\mathbf{0}_T$.
4: **for** $i = 1$ to b **do**
5: MERGE: $\mathcal{M}^{(i)} \leftarrow \big\{(\mathbf{c}, \mathbf{c}^{(i,k)}) : \mathbf{c} \in \mathcal{M}^{(i-1)},\ k \in [S_i]\big\}$ with sums $P_{\mathbf{c}} + \boldsymbol{X}^{(i,k)}$.
6: PARETO-PRUNE: retain $\mathbf{c} \in \mathcal{M}^{(i)}$ only if $\nexists\, \tilde{\mathbf{c}} \in \mathcal{M}^{(i)}$ with $P_{\mathbf{c}} \prec P_{\tilde{\mathbf{c}}}$.
7: ENVELOPE-PRUNE: discard $\mathbf{c} \in \mathcal{M}^{(i)}$ if $\mathcal{L}\big(P_{\mathbf{c}} + \sum_{l=i+1}^{b} \boldsymbol{X}^{(l)}_{\textcircled{\vee}}\big) > \widehat{L}$.
8: Update $\widehat{L} \leftarrow \min_{\mathbf{c} \in \mathcal{M}^{(b)}} \mathcal{L}(P_{\mathbf{c}})$ and output $\mathbf{c}^\star = \arg\min_{\mathbf{c} \in \mathcal{M}^{(b)}} \mathcal{L}(P_{\mathbf{c}})$.

▶ Remark 7 (Correctness of Algorithm 4). For monotonically non-increasing $\mathcal{L}$ and any feasible extension $\mathbf{c}'$ of a partial selection $\mathbf{c}$ on the first i sub-blocks, the partial sum satisfies $P_{\mathbf{c}'} \preceq P_{\mathbf{c}} + \sum_{l=i+1}^{b} \boldsymbol{X}^{(l)}_{\textcircled{\vee}}$ coordinatewise, hence $\mathcal{L}(P_{\mathbf{c}'}) \ge \mathcal{L}\big(P_{\mathbf{c}} + \sum_{l=i+1}^{b} \boldsymbol{X}^{(l)}_{\textcircled{\vee}}\big)$. If this lower bound exceeds the best loss found so far, $\widehat{L}$, no extension of $\mathbf{c}$ can improve on it, and the envelope-pruning step in Algorithm 4 is sound. Together with the Pareto-pruning step it follows that the final selection $\mathbf{c}^\star$ is globally optimal.

A.5 Computing the Gradient

We show as an example the gradient computation for the stabilizing-selection loss and the threshold-linear (disease) loss. We also show that the simplex relaxation of Problem 1 (Section 3.4.1) is convex for the first case, and not convex for the second case.

1. For the stabilizing selection loss $\mathcal{L}(\boldsymbol{X}_{\mathbf{c}}) = \sum_i w_i \boldsymbol{X}_{\mathbf{c}_i}^2$, the relaxed problem is convex in $\mathbf{C}$ (the Hessian is a sum of rank-1 PSD outer products $\mathbf{x}_k \mathbf{x}_k^t$, $\mathbf{x}_k = \mathrm{vec}(\boldsymbol{X}_{\bullet\bullet k})$, weighted by $w_k > 0$), and its gradient and closed-form KKT solution are direct consequences of the quadratic form; the SDP relaxation of Section 3.4 provides a strictly tighter bound and is the route we follow in this paper.
2. For the threshold-linear loss $\mathcal{L} = \sum_i w_i P(D_i = 1|\boldsymbol{X}_{\mathbf{c}})$ with $P(D_i = 1|\boldsymbol{X}_{\mathbf{c}}) = \Phi\big((z_{K_i} - \boldsymbol{X}_{\mathbf{c}_i})/s_i\big)$ and residual standard deviations $s_i \equiv \sqrt{\boldsymbol{\Sigma}^{(\boldsymbol{Y})}_{ii} + \boldsymbol{\Sigma}^{(\boldsymbol{e})}_{ii}} = \sqrt{1 - \boldsymbol{\Sigma}^{(\boldsymbol{X})}_{ii}}$ (using the unit marginal-variance normalization), the gradient is

$$\frac{\partial \mathcal{L}(\mathbf{C})}{\partial C_{ij}} = -\sum_{k=1}^{T} \frac{w_k}{s_k}\, \phi\left(\frac{z_{K_k} - \boldsymbol{X}_{\mathbf{c}_k}}{s_k}\right) X_{ijk}, \tag{7}$$

with matrix form $\nabla\mathcal{L}(\mathbf{C}) = -\boldsymbol{X} \times_3 \Big[\mathbf{w} \odot \boldsymbol{s}^{-1} \odot \phi\big((\boldsymbol{z}_K - \boldsymbol{X}_{\mathbf{c}}) \odot \boldsymbol{s}^{-1}\big)\Big]$, where $\boldsymbol{z}_K = (z_{K_1}, \ldots, z_{K_T})$ and $\boldsymbol{s}_k^{-1} = 1/s_k$. Differentiating once more gives $\partial^2\mathcal{L}/\partial C_{ij}\partial C_{mc} = -\sum_k \beta_k X_{ijk} X_{mck}$ with $\beta_k \equiv w_k (z_{K_k} - \boldsymbol{X}_{\mathbf{c}_k})\, s_k^{-3}\, \phi\big((z_{K_k} - \boldsymbol{X}_{\mathbf{c}_k})/s_k\big)$, whose sign flips as $\boldsymbol{X}_{\mathbf{c}_k}$ crosses z_{K_k}; the Hessian is therefore not positive semidefinite uniformly over $\mathbf{C}$ and the loss is non-convex.

B A Statistical Model from Quantitative Genetics

To put the abstract problem of the previous section in context, we describe a quantitative-genetics model for the joint distribution of the scores and non-score components determining a phenotype across C genomes and T complex traits. The model is general, applying to any organism for which polygenic scores can be computed, and it extends models used in [22, 26] for multiple traits, with two main differences: First, we model explicitly the joint distribution of the individual chromosomes' scores, whereas in [22, 26] a model was given for the entire genomic score. Second, [22, 26] considered embryos derived from the same two parents, yielding a specific genetic relationship matrix $\Sigma^{(C)}$ representing the Identity-by-Descent sharing of siblings, while in our case the genetic relationship matrix may be more general depending on the selection scenario (e.g. unrelated yeast strains or crop varieties, for which $\boldsymbol{\Sigma}^{(\boldsymbol{C})} \approx \boldsymbol{I}_{\boldsymbol{C}}$).

We assume that the genetic architecture of the traits is infinitesimal, namely that there are numerous causal variants, uniformly distributed along the genome. Denote the matrix of quantitative trait values as $\mathbf{Z} \in \mathbb{R}_{C\times T}$, where Z_{ij} denotes the value of trait j for the i-th copy. We can decompose $\mathbf{Z}$ as follows:

$$\mathbf{Z} = \sum_{i=1}^{M} \left(\boldsymbol{X}_{i\bullet\bullet} + \boldsymbol{Y}_{i\bullet\bullet}\right) + \boldsymbol{\epsilon} \tag{8}$$

where the error term $\boldsymbol{Y}$ represents a tensor of genetic components not accounted for by the scores $\boldsymbol{X}$, and the error term $\boldsymbol{\epsilon}$ represents a matrix of environmental components, both having zero mean.

We assume that all the traits have mean zero and variance 1, and further that the individual chromosome scores also have zero mean.

We further assume that the distribution of the polygenic scores $\boldsymbol{X}$ is approximately Normal in each genome (due to the polygenic nature of most complex traits [40]), and that the joint distribution of the polygenic scores over C genomes is multivariate Gaussian.

Consider T traits normalized to have zero mean and unit variance. Let $\boldsymbol{X}^{(e)} \equiv \sum_{i=1}^{M} \boldsymbol{X}_{i\bullet\bullet} \in \mathbb{R}_{C\times T}$ be the entire-genome score matrix for the C copies, obtained by summing the per-chromosome score matrices (superscript e for aggregation over all blocks, as in G_e; not the environmental $\boldsymbol{\Sigma^{(e)}}$), and similarly $\boldsymbol{Y}^{(e)} \equiv \sum_{i=1}^{M} \boldsymbol{Y}_{i\bullet\bullet} \in \mathbb{R}_{C\times T}$. The vector of polygenic score for a single genomic copy for all traits $\boldsymbol{X}_{i\bullet}^{(e)}$ has a covariance matrix $\boldsymbol{\Sigma^{(X)}}$ under a Normal model:

$$\boldsymbol{X}_{i\bullet}^{(e)} \sim MVN(\boldsymbol{0_T}, \boldsymbol{\Sigma^{(X)}}) \tag{9}$$

where $diag(\boldsymbol{\Sigma^{(X)}}) = (\sigma_{11}^{(X)}, .., \sigma_{TT}^{(X)})$ contains the variance explained by the polygenic scores of each trait, and the off-diagonal elements of $\boldsymbol{\Sigma^{(X)}}$ represent pleiotropic effects. For C full-genome copies we obtain a $C \times T$ matrix of polygenic scores with a matrix Normal distribution [19]:

$$\boldsymbol{X}^{(e)} \sim MN_{C\times T}(\boldsymbol{0_{C\times T}}, \boldsymbol{\Sigma^{(C)}}, \boldsymbol{\Sigma^{(X)}}). \tag{10}$$

The matrix $\boldsymbol{\Sigma^{(C)}}$ represents (twice) the kinship coefficients between the C full-genome copies. For example, when the copies represent unrelated strains or lines, $\boldsymbol{\Sigma^{(C)}} = \boldsymbol{I_C}$; when they represent sibling embryos, $\boldsymbol{\Sigma^{(C)}} = \frac{1}{2}[I_C + \boldsymbol{1}_C\boldsymbol{1}_C^t]$. We assume that the chromosome scores are independent, with the scores matrix of each chromosome having the matrix Normal distribution:

$$\boldsymbol{X}_{i\bullet\bullet} \sim MN_{C\times T}(\boldsymbol{0_{C\times T}}, \alpha_i^2\boldsymbol{\Sigma^{(C)}}, \boldsymbol{\Sigma^{(X)}}), \tag{11}$$

where α_i^2 is the proportion of genetic variance explained by chromosome i. The genetic variances satisfy $\sum_{i=1}^{M} \alpha_i^2 = 1$, and this proportion is assumed to be the same for all traits, a consequence of the infinitesimal model and provided that the relative density of causal variants across the genome is similar across traits.

These contributions determine the utility of chromosomal selection, and are expected to be roughly proportional to chromosomes' length or to their number of genes. Here, we show a numerical analysis with α_i^2 taken as the (normalized) chromosome lengths derived from the *S. cerevisiae* reference genome [8] and the human reference genome GRCh38 [37], shown in Table 2. The actual α_i^2 may vary across traits and can be estimated per trait by heritability-partitioning methods [15, 41] when needed.

Table 2 Relative chromosomal lengths (i.e. α_i^2 values) for the yeast (16 chromosomes) and human (22 autosomes plus the X chromosome) reference genomes. The Y chromosome contributes negligibly to autosomal polygenic scores and is omitted. For yeast, $\sum_i \alpha_i \approx 3.89$; for the 23-block human set, $\sum_i \alpha_i \approx 4.68$.

Yeast (*S. cerevisiae*, haploid, $M = 16$)

Chrom.	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI
α_i^2	.019	.067	.026	.126	.047	.022	.090	.046	.036	.061	.055	.089	.076	.064	.090	.078

Human (*H. sapiens*, diploid, $M = 23$)

Chrom.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	X
α_i^2	.082	.080	.066	.063	.060	.056	.053	.048	.046	.044	.045	.044	.038	.035	.034	.030	.028	.027	.019	.021	.015	.017	.052

Similarly, the non-score genetic components are modeled as:

$$\begin{aligned}\boldsymbol{Y}^{(e)} &\sim MN_{C\times T}(\boldsymbol{0_{C\times T}}, \boldsymbol{\Sigma^{(C)}}, \boldsymbol{\Sigma^{(Y)}})\\ \boldsymbol{\epsilon} &\sim MN_{C\times T}(\boldsymbol{0_{C\times T}}, \boldsymbol{I_C}, \boldsymbol{\Sigma^{(e)}})\end{aligned} \tag{12}$$

where $\mathrm{diag}\big(\boldsymbol{\Sigma}^{(\boldsymbol{X})}+\boldsymbol{\Sigma}^{(\boldsymbol{Y})}+\boldsymbol{\Sigma}^{(\boldsymbol{e})}\big)=\mathbf{1}_T$, i.e. each trait is normalized to unit marginal variance while cross-trait covariances may be nonzero. The matrix $\boldsymbol{\Sigma}^{(\boldsymbol{X})}+\boldsymbol{\Sigma}^{(\boldsymbol{Y})}$ is known as the genetic covariance matrix, and can be estimated from GWAS data using e.g. methods like LD-Score-Regression [6]. The diagonal elements $h_i^2=\sigma_{ii}^{(X)}+\sigma_{ii}^{(Y)}$ are the narrow-sense heritabilities of the traits.

The matrix $\boldsymbol{\Sigma}^{(\boldsymbol{Y})}+\boldsymbol{\Sigma}^{(\boldsymbol{e})}$ is the covariance matrix of the residuals, and determines the conditional distribution of the phenotypes vector conditioned on the scores vector. For simplicity, our model makes several standard assumptions: no shared environment (hence the identity $\boldsymbol{I_C}$ is used as a covariance matrix for $\boldsymbol{\epsilon}$), and no assortative mating. If these assumptions are violated, this can be encoded by the covariance matrices of our model.

Threshold traits. For threshold (e.g. disease) traits we use the liability-threshold model [14]. Trait k has prevalence $K_k\in[0,1]$ and liability $z_k=\boldsymbol{X}_{\mathbf{c}_k}+\varepsilon_k$, the selected score plus independent residual noise ε_k of variance $s_k^2=\boldsymbol{\Sigma}_{kk}^{(\boldsymbol{Y})}+\boldsymbol{\Sigma}_{kk}^{(\boldsymbol{e})}=1-\boldsymbol{\Sigma}_{kk}^{(\boldsymbol{X})}$ (the liability variance not explained by the score, with $\boldsymbol{\Sigma}_{kk}^{(\boldsymbol{X})}$ the score's variance-explained on the liability scale). The status is the binary random variable $D_k=\mathbf{1}_{\{z_k<z_{K_k}\}}$ with threshold $z_{K_k}=\Phi^{-1}(K_k)$, and $\mathbf{D}=(D_1,\dots,D_T)$. Conditioning on the score,

$$P(D_k=1\mid\boldsymbol{X}_{\mathbf{c}})=\Phi\big((z_{K_k}-\boldsymbol{X}_{\mathbf{c}_k})/s_k\big),\tag{13}$$

so a large score lowers the crossing probability; the all-pass probability $P(\mathbf{D}=\mathbf{0}_T\mid\boldsymbol{X}_{\mathbf{c}})$ is the Gaussian orthant probability of $\{z_k\geq z_{K_k}\ \forall k\}$ under the residual covariance $\boldsymbol{\Sigma}^{(\boldsymbol{Y})}+\boldsymbol{\Sigma}^{(\boldsymbol{e})}$.

C Implementation Details

Our algorithms and the simulation study presented in the paper are implemented in an R package available at `https://github.com/orzuk/EmbryoSelectionCalculator`, with the chromosomal-selection functions in `code/R/chrom`. Performance-critical routines are implemented in C++ via Rcpp [13].