Uncovering complementary sets of variants for the prediction of quantitative phenotypes

Serhan Yılmaz [1, *], Mohamad Fakhouri [2],
Mehmet Koyutürk [1,3], A. Ercüment Çiçek [2, 4, ‡], and Öznur Taştan [5, §]

[1]Department of Computer and Data Sciences, Case Western Reserve University
[2]Department of Computer Engineering, Bilkent University
[3]Center for Proteomics and Bioinformatics, Case Western Reserve University
[4]Department of Computational Biology, Carnegie Mellon University
[5]Faculty of Engineering and Natural Sciences, Sabanci University

[*]serhan.yilmaz@case.edu, [‡]cicek@cs.bilkent.edu.tr, [§]otastan@sabanciuniv.edu

Abstract

Recent genome-wide association studies (GWAS) show that mutations in single genetic loci, frequently called single nucleotide polymorphisms (SNPs), alone are not sufficient to explain the phenotypic heritability of complex, quantitative phenotypes. Instead, many methods attempt to deal with this issue by considering a set of loci that can characterize the phenotype together. While the state-of-the-art methods are successful in selecting subsets of SNPs that can achieve high phenotype prediction rates, they are either slow in runtime or have hyper-parameters that require further fine tuning through cross-validation or other similar techniques, which makes such methods inconvenient to use. In this work, we propose a fast and simple algorithm named Macarons to select a small, complementary subset of SNPs by avoiding redundant pairs of SNPs that are likely to be in linkage disequilibrium (LD). Our method features two interpretable parameters that control the time/performance trade-off without requiring any hyper-parameter optimization procedures. In our experiments, we benchmark the performance of the SNP selection methods on the 17 flowering time phenotypes of *Arabidopsis Thaliana*. Our results consistently show that Macarons has similar or better phenotype prediction performance while being faster and having a simpler premise than other SNP selection methods.

1 Introduction

A brief overview of GWAS and the SNP selection problem

Genome-Wide Association Studies (GWAS) attempt to find a relation between the genetic variations and a phenotype. In recent studies, many individual variants, also known as single nucleotide polymorphisms (SNPs), have been found to be directly related to various diseases or disorders including type II diabetes, obesity and schizophrenia as well as other quantitative traits like height (Goldstein *et al.*, 2009; Visscher *et al.*, 2017). Yet, the main drawback of focusing on individual SNPs is that they typically fail to explain complex phenotypes (Manolio *et al.*, 2009). Thus, as a more powerful approach, many studies suggest focusing on combinations or sets of SNPs instead of analyzing them individually (Wang *et al.*, 2010; Cordell, 2009; Phillips, 2008; Wei *et al.*, 2014). This study is mainly focused on the problem of finding a subset of SNPs that are *together* predictive of a phenotype. For the sake of brevity, we will simply refer to it as the SNP selection problem throughout this paper.

Approaches investigating combinations of SNPs (e.g epistatis) require limiting the search space

Finding combinations of SNPs that are predictive of a phenotype is computationally challenging due to the sheer number of possible SNP combinations that need to be considered. For this reason, some methods focusing on high-order interactions use exhaustive searches or greedy algorithms on a small, limited pool of SNPs which is usually not more than a few hundreds (Ritchie *et al.*, 2001; Fang *et al.*, 2012). While such pools of "promising SNP candidates" are typically obtained using *a priori* information sources which limit the analysis to coding regions of the genome, it is also possible to do other types of filtering based on automated searches (Ding *et al.*, 2015). Indeed, for combinatorial studies investigating the pair-wise interactions (or as more commonly known as *epistasis*) between the variants in full genome, recent studies show the importance of limiting the search space through prioritization of the tests (Piriyapongsa *et al.*, 2012; Cowman and Koyutürk, 2017), both to alleviate the computational intensity of the task, as well as to improve the overall statistical power. Specifically, a recent study (Caylak and Cicek, 2020) demonstrates the utility of an initial filtering of SNPs based on an automated SNP selection algorithm as a powerful approach that can improve the statistical power considerably.

There are limited options for SNP selection problem in quantitative phenotypes

While there are a number of approaches to choose from for binary phenotypes (e.g., as in case/control studies) (Yu and Liu, 2004; Ding and Peng, 2005; Guestrin *et al.*, 2005; Van Hulse *et al.*, 2012; Ayati and Koyutürk, 2016; Urbanowicz *et al.*, 2018), the options for the SNP selection problem in quantitative phenotypes are more limited. In the machine learning community, this problem essentially translates to a feature selection task for regression analysis and there are many established methods for this purpose. However, due to the high dimensional nature of the problem (hundreds of samples, millions of SNPs), established methods for feature selection such as linear regression with l_1 (lasso) regularization (Tibshirani, 1996; Grave *et al.*, 2011), spectral-relaxation based approaches (Zhao and Liu, 2007; Das *et al.*, 2012), graph-constrained feature selection methods like GraphLasso and GroupLasso (Meier *et al.*, 2008; Jacob *et al.*, 2009), as well as various other methods with sparsity constraints known in the bioinformatics community in the context of other problems (e.g., for selecting gene sets) (Li and Li, 2008; Jia *et al.*, 2011; Liu *et al.*, 2017), are simply too computationally intensive for the SNP selection problem and do not extend to the full scale of variants in the human genome.

To achieve a scalable solution for all known variants in the genome, a new body of SNP selection algorithms have emerged that are designed with computational efficiency in mind (Azencott *et al.*, 2013; Yilmaz *et al.*, 2019). Such algorithms, instead of considering the combinatorial effect of the SNPs on a phenotype, simplify the problem by focusing on a linear combination of individual phenotype associations of SNPs, together with other *a priori* information encoded in the form of a biological network to improve the overall predictivity. In particular, SConES (Azencott *et al.*, 2013) uses a minimum-cut solution under sparsity and connectivity constraints on a SNP-SNP network, and more recently, SPADIS (Yilmaz *et al.*, 2019) uses a SNP-SNP network to enforce the diversity of the selected SNPs.

The drawbacks of existing methods

Linkage disequilibrium (LD) which refers to the non-random association of variants, is a common phenomenon for close variants on the same chromosome (Ardlie *et al.*, 2002). While the connectivity constraint of SConES helps to improve the robustness of the selections, it also promotes the selection of SNPs that are in LD, thereby impairing the phenotype predictions. On the other hand, SPADIS seeks to increase the diversity of SNPs by penalizing the selection of close SNPs on the input network. While this diversity helps to avoid redundant SNPs in LD and improves the phenotype predictions, the main drawback of SPADIS is that it includes two hyperparameters without any interpretable meanings or default values, that need to be optimized through an external procedure such as cross-validation. The need for such external procedures not only makes the method hard to apply from a user viewpoint, but also considerably exacerbates the run time and reduces the robustness of the selections as we demonstrate in our results.

Algorithm 1 Forward Selection Algorithm for the SNP selection problem

Input: Gain function G, ground set V, cardinality constraint $k \leq |V|$.
Output: Set $S \subseteq V$ such that $|S| = k$.

```
S ← ∅
while |S| < k do
    S ← S ∪ argmax_{x∈V\S} G(S, s_x)
end while
```

Macarons: a fast and simple algorithm to select a set of complementary SNPs

To overcome these limitations, we determined three main objectives a SNP selection algorithm should satisfy: (i) Just as predictive as other state-of-the-art SNP selection methods for the prediction of quantitative phenotypes, (ii) Fast enough to consider all variants in the genome as the search space, and (iii) Easy to use without requiring any hyper-parameter optimization procedures. Thus, we propose a new algorithm named Macarons that take into account the correlations between SNPs to avoid the selection of redundant pairs of SNPs in linkage disequilibrium. Overall, Macarons features two simple, interpretable parameters to control the time/performance trade-off: The number of SNPs to be selected (k), and maximum intra-chromosomal distance (D, in base pairs) to reduce the search space for redundant SNPs. Note that, since the parameters have interpretable meanings, they can be determined in advanced (without requiring any procedures for hyper-parameter optimization) with the available computational resources and the goals of further studies in mind.

2 Methods

2.1 Background

Here, we first start by defining the problem as follows: We are given as input a ground set of SNPs V of size n, genotype matrix $\mathbf{X} \in \{0, 1, 2\}^{m \times n}$ decoding the number of alternate alleles for m samples and n SNPs, and a phenotype vector $Y \in \mathbb{R}^{m \times 1}$ containing quantitative values for m samples. Due to the nature of the data, the number of SNPs n is much larger than the number of samples $n \gg m$. Thus, we would like a obtain a small subset of SNPs $S = \{s_1, s_2, \ldots, s_k\}$ of size k that maximizes the prediction performance of the given phenotype vector Y based on a regression model $\mathcal{M}$.

Since this is a computationally intensive problem, state-of-the art methods model the effect of a SNP on the phenotype individually through an association score like sequence kernel association test (SKAT), and aims to combine these using linear models with additional constraints to reduce the overall computational complexity. Specifically, SPADIS uses a greedy algorithm (given in Algorithm 1) that iteratively selects SNPs based on a gain function $\bar{G}(S_t, s_x)$:

$$\bar{G}(S_t, s_x) = c_x + \beta \left(1 - \sum_{s_i \in S_t} K(s_i, s_x) \right) \tag{1}$$

S_t is the subset of selected SNPs at the t iteration of the algorithm, s_x is a candidate SNP being considered, c_x is the phenotype association score of SNP s_x, $K(s_i, s_x)$ is a function that applies a penalization for the pair (s_i, s_x) based on their distance on an input network, and β is an adjustable parameter that is used to convert c_x and $K(s_i, s_x)$ to comparable scales.

2.2 Macarons

Here, we propose a similar approach based on a simpler and more interpretable gain function $G(S_t, s_x)$ that penalizes the selection of highly correlated SNPs:

$$G(S_t, s_x) = c_x \left(1 - R^2(S_t, s_x) \right) \tag{2}$$

where $R^2(S_t, s_x)$ determines the penalization to be applied to a candidate SNP s_x given the selected SNP set $S_t = \{s_1, s_2, \ldots, s_t\}$. Since we would like to penalize SNPs that would be redundant to add, we choose the squared multiple correlation as the penalization function.

This gain function has three main differences to Equation 1: (i) Instead of using shortest-path distance on an input network as a proxy for diversification, it measures correlations between the selected SNPs directly from data, (ii) Instead of a linear formulation, it estimates the overall redundancy of a candidate SNP s_x using multiple correlation coefficient $R^2(S_t, s_x)$, and finally (iii) It uses a multiplicative formula in order avoid using a hyper-parameter like β to bring phenotype association scores and the penalization function to comparable scales.

2.2.1 Estimation of the penalization function

The main issue here with the estimation of the multiple correlation is that it requires the computation of high order interaction terms between the selected SNPs. This makes its estimation for a given data sample both computationally intensive (with $O(mt^2 + t^3)$ runtime complexity), and noisy due to over-fitting to the given sample instances. To help overcome these issues, we first express the calculation of the multiple correlation as a multiplication of several terms involving partial correlations:

$$\begin{aligned} R^2(S_t, s_x) &= 1 - \big(1 - R^2(s_1, s_x|\emptyset)\big)\big(1 - R^2(s_2, s_x|s_1)\big)\ldots\big(1 - R^2(s_t, s_x|s_1, \ldots, s_{t-1})\big) \\ &= 1 - \prod_{i=1}^{t}\big(1 - R^2(s_i, s_x|s_1, \ldots, s_{i-1})\big) \\ &= 1 - \prod_{i=1}^{t}\big(1 - R^2(s_i, s_x|S_{i-1})\big) \end{aligned} \tag{3}$$

where $R^2(s_i, s_j|S')$ denotes the squared partial correlation between SNPs s_i and s_j given the SNPs within the set S'. Note that, these partial correlations are also high-order terms and do not simplify the computation of the multiple correlation by themselves. For this purpose, we make the following simplifying assumption:

$$R^2(s_i, s_x|S_{i-1}) \approx R^2(s_i, s_x|\emptyset) = r^2(s_i, s_x) \tag{4}$$

where $r^2(s_i, s_x)$ is the squared zero-order correlation coefficient (i.e., ordinary Pearson's correlation) between SNPs s_i and s_x. Thus, with this assumption, the estimation of the multiple correlation simplifies to:

$$\bar{R}^2(S_t, s_x) = 1 - \prod_{s_i \in S_t}\big(1 - r^2(s_i, s_x)\big) \tag{5}$$

This assumption both alleviate the overfitting problem in the estimation of multiple correlation (since it overall reduces the number of terms needed to be estimated from the data) as well as reduces the required computation time drastically (from $O(mt^2 + t^3)$ to $O(mt)$).

In the remaining sections of this manuscript, we will refer to the estimation of the squared multiple correlation $\bar{R}^2(S_t, s_x)$ simply as the penalization function, and we will refer to the zero-order correlation $r^2(s_i, s_x)$ as the redundancy function.

2.2.2 Properties of the penalization function

Similar to the squared multiple correlation $R^2(S, s_x)$, the penalization function $\bar{R}^2(S, s_x)$ is always between 0 and 1 and applies diminishing returns principle, obeying the following intuitive properties:

For a given SNP set S and a SNP $s_x \notin S$, the penalization function $\bar{R}^2(S, s_x)$ obeys:

- If S does not include any SNPs correlated with s_x, then the penalization of s_x is zero: i.e., If $r(s_i, s_x)^2 = 0\ \forall i \in S$, then $\bar{R}^2(S, s_x) = 0$.
- If S includes a single SNP s_i, the penalization of s_x is equal to their squared correlation: i.e., if $S = \{s_i\}$, then $\bar{R}^2(S, s_x) = r(s_i, s_x)^2$.

- If S includes a SNP s_i perfectly correlated with s_x, the penalization of s_x is maximum: i.e., if $s_i \in S$ and $r(s_i, s_x)^2 = 1$, then $\bar{R}^2(S, s_x) = 1$ (thus, gain function $G'(S, s_x) = 0$).
- The penalization of s_x is monotonically non-decreasing and can only increase as the set grows: i.e., $\bar{R}^2(S, s_x) \leq \bar{R}^2(S \cup \{s_i\}, s_x)\ \forall s_i \notin S \cup \{s_x\}$.
- Selecting a SNP s_i uncorrelated with s_x does not affect the penalization of s_x: i.e., If $r(s_i, s_x) = 0$, then $\bar{R}^2(S \cup \{s_i\}, s_x) = \bar{R}^2(S, s_x)$.
- If S includes two SNPs s_i and s_j, the penalization of s_x is equal to the sum of individual squared correlations with s_i and s_j minus their expected overlap: i.e., if $S = \{s_i, s_j\}$, then $\bar{R}^2(S, x) = r(s_j, s_x)^2 + r(s_i, s_x)^2 - r(s_j, s_x)^2 r(s_i, s_x)^2$.
- If a new SNP s_i is added to a SNP set S, the joint penalization of s_x is increased by the squared correlation of s_i with s_x minus their expected overlap with S: i.e., If $S' = S \cup \{s_i\}$, then $\bar{R}^2(S', s_x) = \bar{R}^2(S, s_x) + r(s_i, s_x)^2 - \bar{R}^2(S, s_x) r(s_i, s_x)^2$.

2.2.3 Limiting the search space through intra-chromosomal distance

One particular issue for directly using the penalization function given in Equation 5 together with the gain function and algorithm in Equation 2 and Algorithm 1 is that the overall runtime can still be slow for large k (number of SNPs selected) with algorithmic complexity of $O(nmk)$ due to the requirement of computing $O(nk)$ correlation coefficients. For this reason, we make an additional simplifying assumption to limit the search space to intra-chromosomal SNP pairs within a specified distance. Specifically, we assume the following:

$$\left\{ \begin{array}{ll} r^2(s_i, s_j) = 0, & \text{if } s_i \text{ and } s_j \text{ are } \textit{not} \text{ on the same chromosome} \\ r^2(s_i, s_j) = 0, & \text{if } d(s_i, s_j) > D \end{array} \right\} \tag{6}$$

where $d(s_i, s_j)$ is defined as the intra-chromosomal distance between SNPs s_i and s_j (i.e., the absolute difference of their position on the sequence) and D is an adjustable parameter (unit in base pairs) to control the time/performance trade-off of the algorithm by limiting the search space for the redundancy estimations.

2.2.4 Formulation of Macarons algorithm

Overall, with the assumptions given in Equation 4 and Equation 6, the penalization function becomes as follows:

$$\begin{gathered} \bar{r}^2(s_i, s_j) = \left\{ \begin{array}{ll} r^2(s_i, s_j), & \text{if } s_i \text{ and } s_j \text{ are intra-chromosomal with } d(s_i, s_j) \leq D \\ 0, & \text{otherwise} \end{array} \right\} \\ \bar{R}^2(S_t, s_x) = 1 - \prod_{s_i \in S_t} \left(1 - \bar{r}^2(s_i, s_x)\right) \end{gathered} \tag{7}$$

Thus, the gain function becomes:

$$\begin{aligned} G(S_t, s_x) &= c_x \left(1 - \bar{R}^2(S_t, s_x)\right) \\ &= c_x \prod_{s_i \in S_t} \left(1 - \bar{r}^2(s_i, s_x)\right) \end{aligned} \tag{8}$$

The Macarons algorithm that encodes this gain function for step-wise SNP selection is given in Algorithm 2. Overall, it has $O(nk + \lambda_D mk)$ run time complexity where the first term is for maximizing the gain function, and the second term is for computing the gain function (which require the measurement of correlations from data). Here, λ_D is a variable between $[1, n]$ dependent on the D parameter. It represents the average the number of SNP pairs that require the computation of correlation for a given D threshold. Thus, the overall complexity for small D is $O(nk)$ (when the first term dominates) and $O(mnk)$ for large D (as the computation of Pearson correlations becomes the bottleneck).

Algorithm 2 Macarons Algorithm

Input: Ground SNP set V, chromosome numbers and positions for all SNPs $s_i \in V$, the phenotype association scores c_i for all $s_i \in V$, cardinality constraint $k \leq |V|$, trade-off parameter $D \geq 0$ in base pairs

Output: Set $S \subseteq V$ such that $|S| = k$.

$S \leftarrow \emptyset$
$G \leftarrow \{c_i \ \forall i \in V\}$
while $|S| < k$ **do**
  $s_i \leftarrow \underset{s_x \in V \setminus S}{\operatorname{argmax}} G(s_x)$
  $S \leftarrow S \cup \{s_i\}$
  for all intra-chromosomal (s_i, s_x) with $d(s_i, s_x) \leq D$ **do**
    $\bar{r}(s_i, s_x) \leftarrow$ compute Pearson's correlation between s_i and s_x
    $G(s_x) \leftarrow G(s_x)\left(1 - \bar{r}^2(s_i, s_x)\right)$
  end for
end while

3 Results

3.1 Datasets

We use the *Arabidopsis Thaliana* dataset from Atwell *et al.* (2010). This dataset contains 17 phenotypes which are all related to flowering times. There are between 119 and 180 samples (depending on the phenotype). Before applying any filters, the dataset contained 214051 SNPs. SNPs that had a minor allele frequency $< 10\%$ were removed, and 173219 SNP remained. The SNP-SNP network that was used for SPADIS and SConES was constructed by connecting SNPs that are adjacent on the chromosomes. We use two principal components on the genotype data to correct the population stratification (Price *et al.*, 2006) and score each SNP by using the Sequence Kernel Association Test (SKAT) (Wu *et al.*, 2011).

3.2 Background Methods

We compare Macarons with the following methods:

- Baseline: An greedy algorithm that picks the SNPs with the k highest SKAT scores.
- SConES: A SNP selection method that poses the SNP selection problem as a minimum-cut problem (Azencott *et al.*, 2013).
- SPADIS: A SNP selection algorithm that selects distant and complementary set of SNPs Yilmaz *et al.* (2019).

3.3 Experimental Setting

We note that SConES does not directly use the cardinality constraint k, whereas both Macarons and SPADIS do. In order to make the comparison fair, we apply a binary search to find the set of parameters that yield the closest number of selected SNPs during the parameter selection process for SConES. Macarons and SPADIS both to select k SNPs. We repeat the tests for $k = 20, 50, 100, 250, 500, 1000$ to get an idea of how the parameter k affects the methods.

Our testing scheme consists of using a nested cross-validation scheme for each of the 17 phenotypes. We first use 10 cross-validation folds to split the data into train and test data. Then, each outer fold uses 10 inner cross-validation folds for the parameter selection of the methods. After the methods select their parameters, they are tested on the corresponding test set. For each parameter, 6 values using grid-search and chose the one that performed the best was chosen for testing. Since Macarons did not require any hyper-parameter selection, we tested it for two D values. We chose $D = 20$ kbp as suggested by Azencott *et al.* (2013). We also tested $D = \infty$ to be able to cover the entire chromosome.

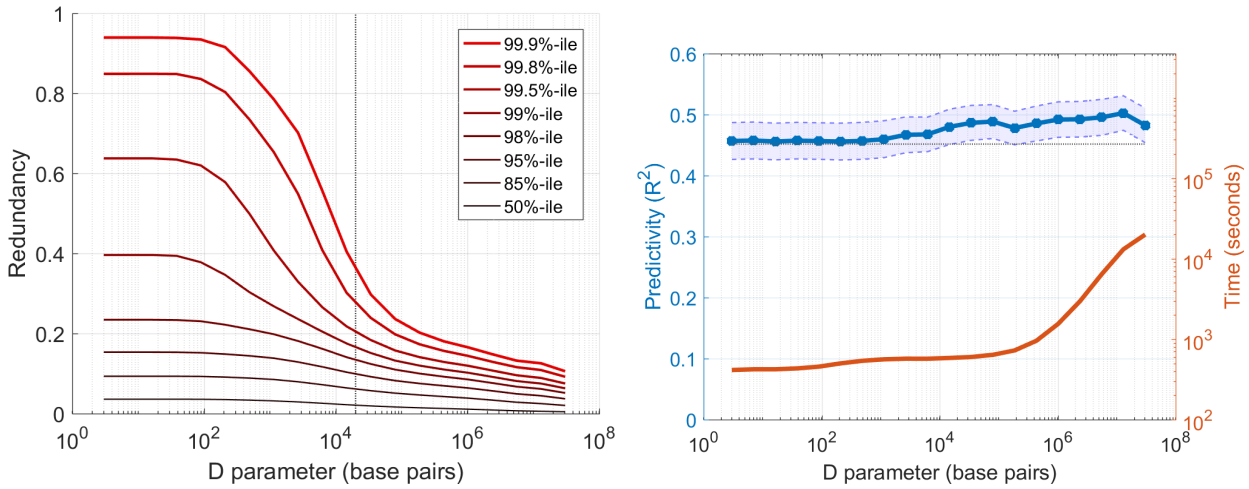

Figure 1: **The predictivity, redundancy, and runtime characteristics of Macarons with respect to its D parameter for K = 1000 selected SNPs.** (Left) Redundancy (measured by squared correlation) between all pairs of selected SNPs. Each line indicates a different percentile for the distribution of redundancy (e.g., if there were 1000 SNP pairs, the 99.5%th percentile would indicate the pair with the 5th highest redundancy value). Note that, the left-most point at the x-axis (for D=1 parameter) corresponds to the baseline method of selecting the highest scoring SNPs (without applying any penalization or regularization). The dotted line indicates *a priori* selected D parameter value of 20 kbps. (Right) The average phenotype prediction performance (Predicitivity, measure by R^2) and the runtime of Macarons with respect to its D parameter. The blue line (left y-axis) shows the average R^2 performance across all 17 phenotypes and the shaded area indicate its 95% confidence interval. The red line (right y-axis) shows the total time required to run Macarons for all phenotypes.

The performance (R^2) was measured using Pearson's squared correlation coefficient. Each method uses ridge regression to set up and train the linear regressor. To choose the ridge regularization parameter, we tested 6 parameters on 5 cross-validation folds.

3.4 Effect of the D parameter

The premise behind Macarons is to select a complimentary set of SNPs while avoiding redundant (correlated) SNPs that are in LD. As we discuss in the methods, the process of taking into account of all redundant SNPs overall requires $k \times n$ (number of selected SNPs $\times$ number of SNPs) correlation estimations from the data, which is both computationally intensive and superfluous since most highly correlated variants tends to be closely located on the genome. To overcome this issue, we limit the search space for correlated SNPs to close intra-chromosomal pairs with maximum distance of D, where D is an adjustable parameter (unit in base pairs).

Here, we investigate the effect of the parameter D for the SNPs selected by Macarons, particularly to examine its effect on the selection of highly correlated SNPs. For this purpose, we select $k = 1000$ SNPs for each of the 17 flowering time phenotype using Macarons with varying D parameters. Then, for each tested value of the D parameter, we investigate the distribution of the redundancy for $\binom{1000}{2} \times 17$ pairs of selected SNPs, in addition to the overall runtime and predictivity characteristics of Macarons (Figure 1). Note that, since the distribution of the redundancy is greatly skewed (i.e., many pairs with low redundancy, a few with very high redundancy), we visualize it using percentile lines (similar to a boxplot) starting from the 50th percentile (median) all the way to the 99.9th percentile. In addition, in Figure 1, right panel, we investigate the effect of D parameter As it can be seen, with *apriori* selected D value of 20 kbps (which is the estimated LD range for *AT* according to Atwell *et al.* (2010)), Macarons can avoid the selection of highly redundant SNPs without making a compromise from the runtime, which translates into an improved phenotype prediction performance.

	Baseline	SConES	SPADIS	Macarons (D=20k)	Macarons (D=Inf)
LN22	0.43	0.42 (-0.1%)	0.44 (+2.0%)	0.43 (+0.3%)	0.42 (-0.1%)
LN16	0.53	0.54 (+1.1%)	0.56 (+3.0%)	0.58 (+5.5%)	0.58 (+4.8%)
LN10	0.44	0.43 (-0.5%)	0.46 (+1.9%)	0.48 (+4.1%)	0.49 (+5.0%)
0W GH LN	0.37	0.37 (+0.2%)	0.37 (+0.3%)	0.39 (+2.6%)	0.35 (-1.8%)
8W GH LN	0.34	0.38 (+3.2%)	0.40 (+5.8%)	0.37 (+2.3%)	0.34 (-0.1%)
SDV	0.38	0.41 (+3.0%)	0.44 (+6.4%)	0.44 (+7.0%)	0.47 (+10.0%)
4W	0.53	0.50 (-2.4%)	0.52 (-1.0%)	0.54 (+1.6%)	0.55 (+2.4%)
8W GH FT	0.39	0.36 (-3.0%)	0.41 (+1.7%)	0.42 (+3.4%)	0.46 (+6.7%)
FT Field	0.52	0.50 (-2.5%)	0.50 (-2.3%)	0.52 (-0.2%)	0.51 (-0.7%)
FRI	0.14	0.12 (-1.8%)	0.11 (-3.6%)	0.13 (-1.7%)	0.13 (-1.6%)
0W GH FT	0.58	0.61 (+2.9%)	0.65 (+6.8%)	0.64 (+6.0%)	0.64 (+6.3%)
FT GH	0.57	0.56 (-0.8%)	0.60 (+2.7%)	0.60 (+2.9%)	0.64 (+6.6%)
SD	0.56	0.55 (-0.5%)	0.54 (-1.1%)	0.58 (+2.0%)	0.62 (+6.8%)
0W	0.51	0.54 (+3.2%)	0.52 (+1.0%)	0.51 (+0.0%)	0.52 (+0.9%)
LDV	0.59	0.59 (+0.4%)	0.62 (+2.5%)	0.62 (+3.2%)	0.60 (+1.3%)
FLC	0.23	0.24 (+1.2%)	0.22 (-0.7%)	0.26 (+3.1%)	0.26 (+3.1%)
2W	0.58	0.59 (+0.7%)	0.67 (+8.3%)	0.63 (+4.9%)	0.68 (+10.1%)

Color Legend: +20% … 0 … -20%

Figure 2: **The phenotype prediction performances of the SNP selection methods compared to the baseline method for $k = 1000$ selected SNPs.** The y-axis corresponds to the 17 flowering time phenotypes and the x-axis corresponds to the SNP selection methods. The leftmost one is a simple baseline method that selects the highest scoring SNPs without any penalization or regularization. Each cell contains two values: (i) The phenotype prediction performance (R^2) for the corresponding method and the phenotype, and (ii) the value in parenthesis shows the difference in R^2 compared to the baseline method as a percentage. Each cell is colored according to their relative improvement compared to baseline: Positive values (indicating improving) are colored blue, and the negative values (indicating decline in R^2 performance) are shown with red colors.

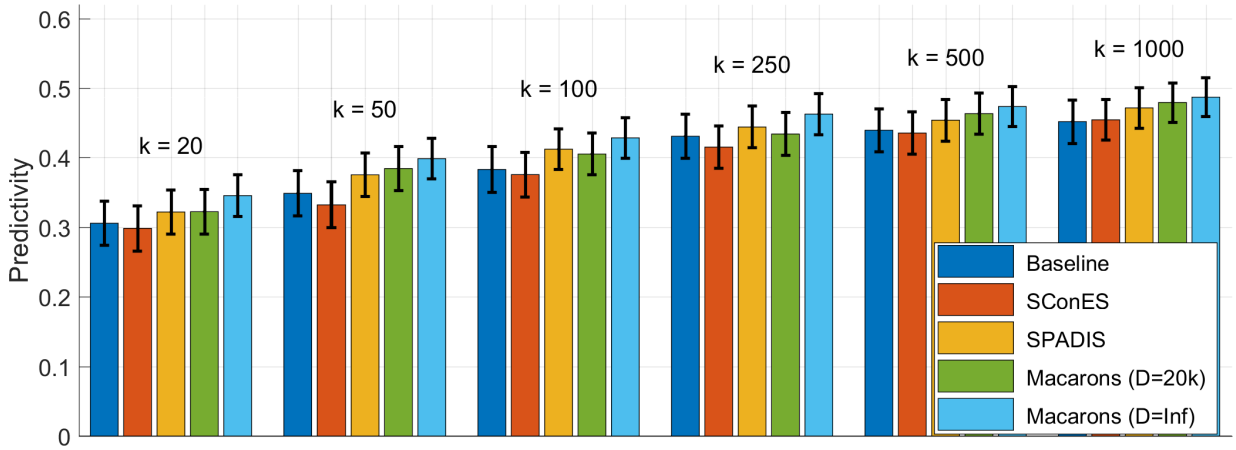

Figure 3: **The phenotype prediction performances of the SNP selection methods averaged across all flowering time phenotypes for different number of selected SNPs (indicated by k values).** The methods are tested for $k = 20, 50, 100, 250,$ and 1000 selected SNPs. Each colored bar represents a different SNP selection method. The y-axis shows the averaged Predictivity (measured by Pearson's squared correlation coefficient, R^2) across all 17 flowering time phenotypes. The black lines indicate the 95% confidence interval for the average R^2 performance for the corresponding method and the k value.

3.5 Benchmarking the predictivity of SNP selection methods

We tested each method on every phenotype for each of the six k values. The results for $k = 1000$ are shown in Figure 2. Here, we see that Macarons with $D = 20$ kbp performs better than the baseline in 15 out of the 17 phenotypes. On the other hand, Macarons with $D = \infty$ performs better than the baseline in 12 out of the 17 phenotypes. On 11 of the 17 phenotypes, using Macarons with $D = \infty$ instead of with $D = 20$ kbp yields better predictive performance. Macarons with $D = 20$ kbp and $D = \infty$ perform better than SPADIS in 12 and 11 phenotypes respectively. Compared to SConES, Macarons with $D = 20$ kbp and $D = \infty$ perform better in 15 and 14 phenotypes respectively. Overall, compared to the other methods, Macarons with $D = 20$ kbp performs better in than the baseline the most phenotypes (15 / 17). We now look at the best performance per phenotype across each methods. SConES achieves the best performance across all methods in only one phenotype. Similarly, SPADIS achieves the best performance on 3 phenotypes. Macarons with $D = 20$ kbp achieves the maximum performance on 5 phenotypes. Macarons with $D = \infty$ takes the win: it achieves the highest performance across 9 phenotypes. Macarons with $D = 20$ kbp and $D = \infty$ are both tied for the highest performance in one phenotype. From this perspective, we see that Macarons gives the best results per phenotype between all the methods.

In Figure 3, we consider the averaged predictivity performances across all phenotypes. Here, we see that as we increase the number of selected SNPs, the overall performances of all methods increases. This is expected since choosing more SNPs gives more information. In this context, Macarons with $D = 20$ kbp and $D = \infty$ perform better than the baseline and SConES across all k values. Macarons with $D = 20$ kbp performs better than SPADIS in $k = 20, 50, 500, 1000$. Meanwhile, Macarons with $D = \infty$ performs better than all methods for all k values.

3.6 Benchmarking other characteristics of SNP selection methods

We define two new metrics here: redundancy and robustness. Redundancy is simply how related two SNPs are, and is measured by Pearson's square coefficient. To investigate the presence of highly correlated SNPs, we calculate the squared correlation between all pairs of selected SNPs, and use the 99.9th percentile (in a similar manner to Figure 1) as the redundancy estimation for the corresponding method. Lower redundancy score means that the method chooses SNPs that do not correlate with, or explain, each other highly. We focus on all phenotypes for $k = 1000$ in Figure 4. We find that the baseline and SConES have the highest redundancy scores. On the other hand, Macarons with $D = \infty$ has the least redundancy score. This is to be expected as increasing the range parameter D means that the method can search over a bigger set. Macarons with $D = 20$ kbp and SPADIS are similar in their redundancy performances.

Robustness metric explains how consist each method is in choosing the same SNPs over different cross-validation folds. This is measured for each method by the averaged Jaccard index between all pairs of folds (each method in each fold has a selected SNP subset). We find that baseline is the most robust, followed by Macarons with $D = 20$ kbp. Macarons with $D = \infty$ on the other hand is the least robust - which is also not surprising.

Predictivity explains how good a SNP subset is for the prediction problem and is measured by Pearson's squared coefficient (R^2). In Figure 4, this is calculated as the average predictivity score across all phenotypes for $k = 1000$. We see that baseline performs the worst. Macarons with $D = 20$ kbp and $D = \infty$ are the highest two performing methods, and SPADIS is in-between.

For time, we measure the total time required (in seconds) to run the corresponding method for all phenotypes on a 40 core machine (each core is Intel(R) Xeon(R) CPU E5-2650 v3 2.30GHz). The experiment was run on 17 cores (one for each phenotype). We see that the baseline is the fastest, followed by Macarons with $D = 20$ kbp. SConES, SPADIS, and Macarons with $D = \infty$ are all relatively slower compared to baseline and Macarons with $D = 20$ kbp. SConES and SPADIS are slow because of the time required to optimize their hyper-parameters, and Macarons with $D = \infty$ is slow due to the large search space.

Acknowledgements

The authors declare that they have no competing interests.

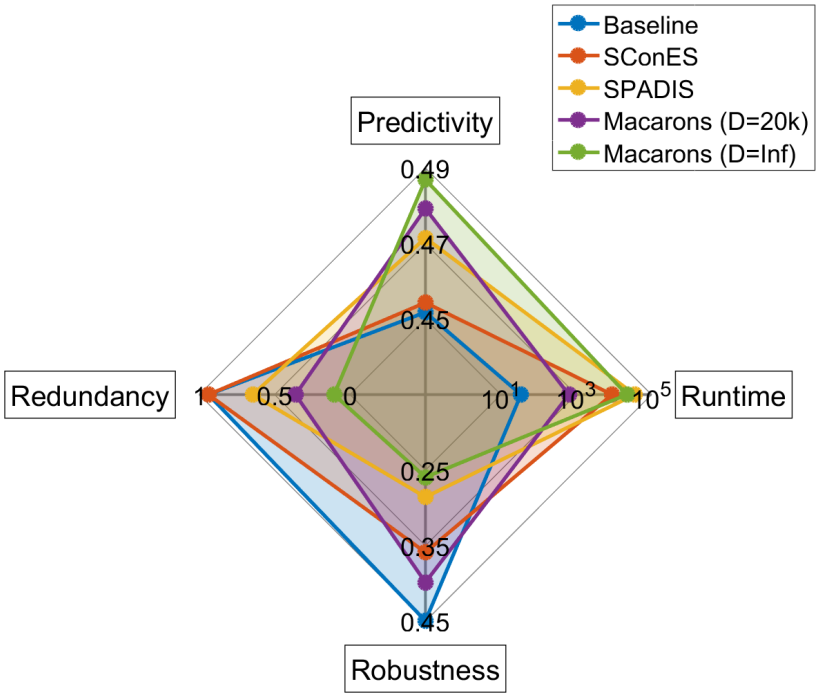

Figure 4: **Top-level overview of the characteristics of the SNP selection methods in terms of their predictivity, redundancy, runtime, and robustness.** The predictivity indicate the average phenotype prediction performance (measured by Pearson's square coefficient, R^2) of the corresponding method for $k = 1000$ selected SNPs. The redundancy axis indicates the presence of high correlation among the selected SNPs (measured by 99.9th percentile of the squared correlation between all pairs of selected SNPs) The robustness axis indicates the consistency of the selected SNPs in different cross-validation folds (measured by averaged Jaccard Index between all pairs of folds). The time axis (in log-scale) shows the total time required (in seconds) to run the corresponding method for all 17 phenotypes in *AT* dataset.

References

Ardlie, K. G. *et al.* (2002). Patterns of linkage disequilibrium in the human genome. *Nature Reviews Genetics*, **3**(4), 299–309.

Atwell, S. *et al.* (2010). Genome-wide association study of 107 phenotypes in arabidopsis thaliana inbred lines. *Nature*, **465**(7298), 627–631.

Ayati, M. and Koyutürk, M. (2016). Pocos: Population covering locus sets for risk assessment in complex diseases. *PLoS computational biology*, **12**(11), e1005195.

Azencott, C.-A. *et al.* (2013). Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics*, **29**(13), i171–i179.

Caylak, G. and Cicek, A. E. (2020). Potpourri: An epistasis test prioritization algorithm via diverse snp selection. In *International Conference on Research in Computational Molecular Biology*, pages 243–244. Springer.

Cordell, H. J. (2009). Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, **10**(6), 392–404.

Cowman, T. and Koyutürk, M. (2017). Prioritizing tests of epistasis through hierarchical representation of genomic redundancies. *Nucleic acids research*, **45**(14), e131–e131.

Das, A. *et al.* (2012). Selecting diverse features via spectral regularization. In *Advances in neural information processing systems*, pages 1583–1591.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, **03**(02), 185–205.

Ding, X. *et al.* (2015). Searching high-order snp combinations for complex diseases based on energy distribution difference. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, **12**(3), 695–704.

Fang, G. *et al.* (2012). High-order snp combinations associated with complex diseases: efficient discovery, statistical power and functional interactions. *PloS one*, **7**(4), e33531.

Goldstein, D. B. *et al.* (2009). Common genetic variation and human traits. *New England Journal of Medicine*, **360**(17), 1696.

Grave, E. *et al.* (2011). Trace lasso: a trace norm regularization for correlated designs. *Advances in Neural Information Processing Systems*, **24**, 2187–2195.

Guestrin, C. *et al.* (2005). Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 265–272, New York, NY, USA. Association for Computing Machinery.

Jacob, L. *et al.* (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM.

Jia, P. *et al.* (2011). dmgwas: dense module searching for genome-wide association studies in protein–protein interaction networks. *Bioinformatics*, **27**(1), 95–102.

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, **24**(9), 1175–1182.

Liu, Y. *et al.* (2017). Sigmod: an exact and efficient method to identify a strongly interconnected disease-associated module in a gene network. *Bioinformatics*, **33**(10), 1536–1544.

Manolio, T. A. *et al.* (2009). Finding the missing heritability of complex diseases. *Nature*, **461**(7265), 747–753.

Meier, L. *et al.* (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**(1), 53–71.

Phillips, P. C. (2008). Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, **9**(11), 855–867.

Piriyapongsa, J. *et al.* (2012). iloci: a snp interaction prioritization technique for detecting epistasis in genome-wide association studies. In *BMC genomics*, volume 13, page S2. Springer.

Price, A. L. *et al.* (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, **38**(8), 904–909.

Ritchie, M. D. *et al.* (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics*, **69**(1), 138–147.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1), 267–288.

Urbanowicz, R. J. *et al.* (2018). Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of biomedical informatics*, **85**, 168–188.

Van Hulse, J. *et al.* (2012). Threshold-based feature selection techniques for high-dimensional bioinformatics data. *Network modeling analysis in health informatics and bioinformatics*, **1**(1-2), 47–61.

Visscher, P. M. *et al.* (2017). 10 years of gwas discovery: biology, function, and translation. *The American Journal of Human Genetics*, **101**(1), 5–22.

Wang, Z. *et al.* (2010). A general model for multilocus epistatic interactions in case-control studies. *PLoS One*, **5**(8), e11384.

Wei, W.-H. *et al.* (2014). Detecting epistasis in human complex traits. *Nature Reviews Genetics*, **15**(11), 722–733.

Wu, M. C. *et al.* (2011). Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, **89**(1), 82–93.

Yilmaz, S. *et al.* (2019). Spadis: An algorithm for selecting predictive and diverse snps in gwas. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1.

Yu, L. and Liu, H. (2004). Redundancy based feature selection for microarray data.

Zhao, Z. and Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157.