Chapter 31
Evolving SNP Panels for Genomic Prediction

Ian Whalen, Wolfgang Banzhaf, Hawlader A. Al Mamun and Cedric Gondro

Abstract The use of genetic variation (DNA markers) has become widespread for prediction of genetic merit in animal and plant breeding and it is gaining momentum as a prognostic tool for propensity to disease in human medicine. Although conceptually straightforward, genomic prediction is a very challenging problem. Genotyping organisms and recording phenotypic traits are time consuming and expensive. Resultant datasets often have many more features (markers) than samples (organisms). Therefore, models attempting to estimate the effects of markers often suffer from overfitting due to the curse of dimensionality. Feature selection is desirable in this setting to remove markers that do not appreciably affect the trait being predicted and amount to statistical noise. We present a differential evolution system for feature selection in genomic prediction problems and demonstrate its performance on simulated data. Code is available at: https://github.com/ianwhale/tblup.

Key words: differential evolution, evolutionary computation, genomic prediction, feature selection

Ian Whalen
Beacon Center for the Study of Evolution in Action and Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA e-mail: whalenia@msu.edu

Wolfgang Banzhaf
Beacon Center for the Study of Evolution in Action and Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA e-mail: banzhafw@msu.edu

Hawlader A. Al Mamun
CSIRO Data61, Commonwealth Scientific and Industrial Research Organisation, Canberra, ACT, Australia e-mail: hawlader.almamun@data61.csiro.au

Cedric Gondro
Beacon Center for the Study of Evolution in Action and Department of Animal Science, Michigan State University, East Lansing, MI, USA e-mail: gondroce@msu.edu

31.1 Introduction

The use of DNA markers for prediction of genetic merit has become widespread in plant and animal breeding and is gaining momentum as a prognostic tool for susceptibility to disease in human medicine. Meuwissen, Hayes, and Goddard [39] introduced the idea of using a very large number of genotypic markers to predict phenotypes. This process is known as genomic prediction and tasks a system with estimating the joint effects of thousands of markers, usually single nucleotide polymorphisms (SNP) on a trait. For agricultural applications, these estimated SNP effects are then used to predict phenotypes or breeding values for new individuals that do not have trait information but do have genotype (marker) information. Over the past ten years, genomic prediction has been widely adopted in genomic selection [21] in agriculture [**?**, 36] and in human studies [1]. Hayes, Bowman, Chamberlain, and Goddard [27] emphasize its value, touting genomic selection as the most significant advancement for the dairy industry in the last two decades.

Although conceptually straightforward, genomic prediction is a very challenging problem. Genotyping and trait recording are costly and time demanding exercises; the result is that most genomic datasets will have hundreds of thousands or even millions of markers for which effects need to be simultaneously estimated from usually only a few thousand phenotyped individuals. This means that the datasets are underdetermined (also known as the $p \gg n$ problem) and suffer from overfitting due to the curse of dimensionality. In effect, genomic prediction can be treated as a high dimensionality, sparse data problem and, consequently, suffers from the same issues as other problems in this domain. Most notably being that the prediction models derived by statistical inference are sub-optimal since the accuracy of the parameter estimates (marker effects) rapidly decays as the number of features that needs to be estimated increases. The accuracy of prediction is also conditional on the genetic architecture of the traits – the interplay between genotypes and phenotypes is complex and varies widely from trait to trait; e.g. highly heritable traits regulated by a few genes of large effect are easier to predict than traits regulated by many genes with small effects and with a low heritability [28]. Moreover, there are still various other factors that will also influence the accuracy of prediction such as marker density (if the data is not at full sequence resolution), the effective population size (N_e), measures of linkage disequilibrium and family relationships [9, 20, 54], population stratification [40], sample size, reliability of phenotypes [19], and the methodology used to estimate marker effects [9].

For these reasons, genomic datasets are prime candidates for feature selection techniques. However, popular methods for genomic feature selection are often statistically-based; e.g. genome wide association studies (GWAS) which aim to identify, in the case of sequence data, the causal variants of a given trait or, when SNP arrays are used, the markers that are in high linkage disequilibrium with the causal variants [29, 52]. These approaches are limited to local searches of the feature space since they are conditioned on the supporting statistical evidence. On the other end of the spectrum, all markers are simultaneously used for prediction irre-

spective of them having or not a functional role on the trait – this is the main method currently adopted for genomic prediction.

Even though quantitative traits are largely polygenic with hundreds or thousands of variants influencing a trait, it still stands to reason that not every single genetic variant across the genome will have a real effect on every single trait. This suggests that current methods lead to sub-optimal accuracy of genomic prediction, especially with sequence data, due to background noise introduced by the large number of spurious non-causative variants included in the prediction models. Under this rationale, we suggest that better prediction models are attainable by using only subsets of markers that are truly informative of a given trait. In this paper we suggest that genomic prediction should be treated as a feature selection problem and that it is amenable to non-statistical methods since they are potentially better at performing global searches of the feature space. Herein we discuss a non-statistical approach for genomic prediction through the use of an evolutionary computation (EC) technique called differential evolution (DE) [49] and compare its performance to mainstream methods.

31.2 Previous Work

31.2.1 Genomic Prediction

Genomic prediction is the process of using a large number of genetic markers to predict phenotypic traits [39]. There are two main approaches used to estimate marker effects. The first approximates a traditional infinitesimal model that assumes all markers—usually single nucleotide polymorphisms (SNP)—contribute a non-zero value to the genetic variance and that SNP effects are normally distributed. The second approach is based on nonlinear methods that emphasize certain genomic regions and allow marker effects to come from distributions other than a Gaussian.

Linear methods like ridge regression best linear unbiased prediction (RRBLUP) [53] and genomic best linear unbiased prediction (GBLUP) [25, 51] follow the assumption that all markers have some nonzero, normally distributed effect[1]. Such methods are well-studied and have been applied across many domains [50, 56]. We point out that all linear methods like the ones mentioned share a common flaw. Assuming that all markers contribute a nonzero effect leads to loci that do not affect the output being assigned an effect value. This amounts to the linear model fitting to statistical noise, which will be detrimental to performance. Thus, feature selection is desirable in order to remove these non-informative marker sites.

The non-linear methods for genomic prediction include Bayes A, Bayes B [39], Bayes C [26], Bayesian Lasso [11], and Bayes R [15]. These methods mainly differ in their assumptions about what distributions the marker effects should follow. Even though a large proportion of the variants might be allocated to a distribution with

[1] These methods are equivalent [24].

very small to zero effects, these methods will still assign a nonzero posterior density to most variants. Hence the number of variants to be used for prediction is still very large and, in the same manner as the linear methods, the Bayesian methods also have limited discrimination between markers with and without an actual effect.

Which of these methods performs better depends on the underlying genetic architecture of the trait, e.g. Bayesian methods tend to outperform BLUP approaches when the trait is less polygenic. In practice, differences in prediction accuracy between methods have been generally very small. While these methods have well characterized statistical properties they are constrained by the underlying model assumptions and, given the dimensionality of the solution space, even very small estimates of effects in non-informative markers will, collectively, reduce prediction accuracy. This is an increasing problem with the increasing number of genetic variants to predict from.

31.2.2 Feature Selection

Feature selection is a subdiscipline of a larger class of techniques known as dimensionality reduction, a well studied problem in machine learning. Feature selection seeks to retain a subset of the original set of features, rather than transform them in some way. This can be preferable to feature transformation—which constructs some function of all input features—since it is interpretable and can lead to deeper understanding about what markers affect a trait most significantly. Genomic data often has the pervasive quality of orders of magnitude more features than samples ($p \gg n$ problem), making it a natural candidate for dimensionality reduction. However, work has been relatively limited to statistical filtering [47]. Here, we present relevant filter and wrapper methods.

Filter methods are often used as a preprocessing step in machine learning problems, functioning independently of any actual modeling. Features are selected based on some statistical relationship with the predicted output and possibly other features. Such examples are univariate correlation significance values or redundancy [7, 43]. Filter methods work by suppressing the least interesting variables, leaving the more promising variables to be used to learn a predictive model. Filter methods tend to be quite computationally efficient and are robust to overfitting, making them a popular choice for genomic classification [3, 8, 12]. To a large extent a genome-wide association study can be viewed as a filtering method – in its simplest form, a GWAS is just a univariate correlation on each SNP that calls attention to significant markers in the genome and is a standard technique to identify genomic regions of interest for a trait [29, 52].

Wrapper methods are of particular interest here because they include EC. These methods iteratively update a feature subset over time, preferring those that perform better according to some measure. Classically, EC is an effective population-based heuristic search inspired by biological principles [14]. Evolutionary computation includes a diverse catalog of methods like genetic algorithms (GAs) [31, 22], genetic

programming [35, 4], evolutionary strategies [46, 48], and ant colony optimization [13]. Storn and Price [49] introduced DE as a greedier alternative to genetic algorithms and evolutionary strategies. There are common themes throughout all EC methods. A group of *individuals* that each represent a solution to some problem have a *fitness* assigned to them based on an objective function. In general, EC has the luxury of being able to use almost any objective function—non-differentiable or otherwise—due to its gradient-free nature. Individuals then combine—or share information—with each other. Those with more desirable fitness values are *selected* to remain in the population and guide a search toward the global optimum.

Feature selection with EC has been applied to a variety of domains problems. Raymer et al. demonstrated the efficacy of dimensionality reduction with a GA on a protein water-binding site identification problem, showing better performance than sequential feature selection methods [45]. Firpi and Goodman showed in [17] that particle swarm optimization can also produce similar results to a GA in multiple applications. Luque-Baena, et al. [37] showed that a simple GA could outperform the state of the art on a cancer pathway identification and classification task. Furthermore, it was shown in [38] that a GA outperforms simple sequential feature selection methods. Feature subsets discovered in [38] were also deemed more biologically relevant, potentially furthering the understanding about the traits being predicted.

Feature selection using DE is relatively new, with first successes being shown in 2008 [33]. In that work, DE was shown to outperform other wrapper methods like particle swarm optimization and genetic algorithms in an electroencephalogram classification task. The method presented here is the technique in [2, 16]. Both works deal with using DE to do feature selection in cattle applications. The original applications performed well compared with random search in [16] and GBLUP in [2]. We further contribute here by the addition of methods to control overfitting.

31.3 Methods

31.3.1 Differential Evolution

Differential evolution is a real-valued optimization technique originally introduced by Storn and Price [49]. The strategy for DE introduced here is the original implementation. See Algorithm 1 for an overview which is abstracted into four main parts: evaluation, mutation, crossover, and selection. We present the relevant descriptions of each of these operations here.

The foundation of DE is a population of n candidate solutions, all of which are vectors in $\mathbb{R}^d$, where d is the dimensionality of the given problem. Each solution, $\mathbf{X}_i = [x_{1,i}, \ldots, x_{d,i}]$ has an associated fitness, which is some performance metric to be maximized. At the beginning of the search, each vector is randomly initialized according to a uniform random distribution, so that $0 \leq x_{j,i} < 1, \forall j$.

Evaluation

The least complex operation in DE is evaluation. Evaluation simply assigns each vector in the population a fitness based on an objective function. See Section 31.3.2 for specifics on how a feature subset is extracted from a real vector and assigned a fitness.

Mutation

As noted, we use the original mutation rule presented in [49]. Known as DE/rand/1, the rule uses a *mutation factor* hyperparameter $F > 0$. A *donor vector*, $\mathbf{V}_i$, is created for each vector in the population, $\mathbf{X}_i$, according to the following equation

$$\mathbf{V}_i = \mathbf{X}_a + F \cdot (\mathbf{X}_b - \mathbf{X}_c)\,. \tag{31.1}$$

Where a, b, c are unique random integers from 1 to population size n[2].

Crossover

Also known as parameter mixing, crossover combines information across solutions. Again, we use the method in [49] known as uniform—or binomial—crossover. For each index j in $\mathbf{X}_i$ and $\mathbf{V}_i$, the following is applied to create *trial vector* $\mathbf{U}_i$

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } rand[0,1) < C_r \text{ or } j = j_{rand} \\ x_{j,i} & otherwise. \end{cases} \tag{31.2}$$

Here, $C_r \in [0,1]$ is the crossover rate hyperparameter, $rand[0,1)$ is a uniform random number in the half-open range $[0,1)$, and j_{rand} is a uniform random integer from $[1,d]$ that is generated once per generation for each solution in the population. The purpose of j_{rand} is to ensure at least one index is crossed-over with the donor vector for each vector in the population.

Selection

The standard selection operator at each generation is a simple tournament selection between $\mathbf{X}_i$ and $\mathbf{U}_i$. If the fitness of $\mathbf{U}_i$ is greater than the fitness of $\mathbf{X}_i$, it replaces $\mathbf{X}_i$ and continues on to the next generation in the i^{th} index of the population.

Note that more recent methods combine DE with other heuristics [6, 34] or use more sophisticated update rules [32, 44]. However, we prefer the original imple-

[2] Storn and Price originally named this update rule as a mutation [49]. For algorithms like GAs and genetic programming, a mutation carries out changes on a single individual in the population—rather than the multiple shown in Equation 31.1.

Algorithm 1 Differential Evolution

Input: Population size n, dimensionality d, generations g
Output: Solution P_{best}

```
P = {X_i | X_i ∈ [0,1)^d, 1 ≤ i ≤ n}
evaluate(P)
for 1 to g do
    P' = {}
    for i = 1 to n do
        V_i = mutate(X_i)
        U_i = crossover(V_i, X_i)
        P' = P' ∪ {U_i}
    end for
    evaluate(P')
    P = selection(P, P')
end for
return P_best
```

mentation for its generality and success across a variety of domains as noted in [10].

31.3.2 Random Keys

Differential evolution in its standard form is a real valued optimization technique. Therefore, some accommodation must be made to obtain indices of a feature subset from a vector of real numbers. Here, we use a technique called *random keys* [5]. The random key technique is well known in EC due to its use in combinatorial optimization tasks such as scheduling [41]. Random keys represent solutions to a combinatorial problem as a real valued vector that is somehow decoded to produce a solution that is always valid in the objective space. This is in contrast to traditional binary encoding that—when acted on by operators like mutation and crossover—may no longer be a valid solution (e.g., a solution selects more than the desired number of features after crossover).

We demonstrate the random key decoding process used for feature selection with an example. Consider a scenario with a five dimensional data set. Each solution in the DE population is then a vector in $\mathbb{R}^5$. The vector

$$[0.08, 0.53, 0.91, 0.34, 0.18].$$

decodes to the feature ordering

$$[3, 2, 4, 5, 1].$$

The feature ordering is obtained by finding indices of the original solution vector in sorted order. More explicitly, observe that 0.91 is at index 3 in the example. Hence, 3 is the first value of the decoding since 0.91 is the largest value in the vector, and so on. If the task was to select two features from the original five, the features at indices 3 and 2 would be selected. Therefore, over time, the DE search will tend to increase the values in the solution vector at indices that tend to increase fitness. A fitness function is then applied to the obtained feature subset. In our case, this is the absolute value of Pearson's correlation between predicted and true phenotypes using RRBLUP [53].

31.3.3 Self-adaptive Differential Evolution

The above description of DE leaves out discussion on tuning the associated hyperparameters F, C_r, and N_p. These values often have a dramatic influence on the convergence of DE [32, 44]. As a result, some effort has gone into alleviating the choice of F and C_r through *self-adaptive* methods that "learn" these values throughout the course of a DE experiment. In the method presented below, this is done by observing which particular settings create trial vectors that successfully enter the next population.

Qin and Suganthan present Self-adaptive Differential Evolution (SaDE) in [44] as a way to learn not only the F and C_r parameters, but which mutation method to use as well. For each individual, a donor vector has probability p of being created with DE/rand/1 and probability $1-p$ of being created with DE/current-to-best/1. Where p is initialized to be 0.5 and updated by calculating

$$p = \frac{ns_1 \cdot (ns_2 + nf_2)}{ns_1 \cdot (ns_2 + nf_2) + ns_2 \cdot (ns_1 + nf_1)}. \tag{31.3}$$

Where ns_1 and nf_1 are the number of trial vectors that were produced with donor vectors from DE/rand/1 that entered the next population (a *success*) and the number that did not (a *failure*), respectively. The values ns_2 and nf_2 have the same definition, but count the number of successes and failures for DE/current-to-best/1. Finally, the first 50 generations of the search do not update p to allow some time for the algorithm to stabilize and learn meaningful success and failure rates [44].

Values of F and C_r are newly generated for each individual solution vector at each generation. F is not learned using any particular scheme in SaDE, and is simply randomly sampled from the normal distribution $\mathcal{N}(0.5, 0.3^2)$, then clipped to fall in the range $(0, 2]$. The authors state that C_r is much more important to the performance of DE and chose to adjust it based on the trajectory of the search [44]. To do so, a C_r is sampled for each index in the population every 5 generations from $\mathcal{N}(C_{rm}, 0.1^2)$. Then, similarly to the mutation strategy, every 25 generations, C_{rm} is recalculated based on the values of C_r that successfully produced trial vectors that entered the

next population. This method has proved successful on many test problems, so it will be applied here as well.

31.3.4 Seeded Initial Population

In order to incorporate domain knowledge, the results of a GWAS can be included in the initial DE population. Through *seeding*, the indices corresponding to the s most significant SNPs are marked with a value of 1 in some vector in the population. Since all solution vectors are initialized in the range $[0,1)$, these indices will form the subset for that particular vector. Due to the greedy nature of DE, the search will never reach a fitness value that is worse than the seeded initial vector.

31.3.5 Heritability Thresholding

The concept of heritability is well known to geneticists studying genomic prediction. At a high level, heritability[3] is the proportion of variance in a phenotype that can be explained only by effects in the genotype. Heritability, h^2 is quite useful since it can be used to define h (i.e., $\sqrt{h^2}$), which is the theoretical limit on prediction for a genomic prediction task. In practice, heritability can be estimated for a trait by analyzing the behavior of a trait through familial lines using a restricted maximum likelihood technique [18, 42].

Reaching values equal to or greater than h means the search has begun to overfit to the validation set since it has gone above the highest possible accuracy. The peculiarity of h should be emphasized. It is quite uncommon for any predictive modeling problem to have a hard threshold for performance. Usually a practitioner shoots for some value that is as high as possible. The method explored here is an initial look into using this value for preventing overfitting in search based feature subset selection.

A rudimentary method is to simply stop the search when some statistic of the population reaches $h(1+\alpha)$ for some small $\alpha \in (-1,1)$. Where these possible measures could be the maximum, minimum, median, or mean fitness of the population. This method is based on the fact that we can treat h as a hard threshold and the idea that it could be better to simply stop searching rather than continue a search that is already known to be overfit to the validation set.

[3] For this discussion, heritability is limited to "narrow-sense" heritability which is captured by additive affects of alleles in a genotype [23].

31.4 Data

31.4.1 Simulation

To demonstrate the performance of DE, simulated data is favorable to control the complexities introduced in genomic data. Specifically, the number of QTL, desired trait heritability, h^2_{in}, and absence of epistatic effects are controlled for through the following process. For this study, $h^2_{in} = 0.4$ Using the genotype matrix $\mathscr{X} \in \{0,1,2\}^{n \times d}$, q columns are uniformly chosen the QTL. Let $\beta^* = [\beta_1, \ldots, \beta_q]^T$ be the true, simulated QTL effects. For this study $q = 100$. First, $\beta_i \sim \mathscr{N}(0,1),\ \forall i$. Then, β^* is adjusted by calculating the variance of each allele. For diploid organisms, there are three genotypes: heterozygous (AB) and homozygous (AA, BB). To determine the genetic variance, we calculate the rate that each genotype occurs (i.e. total occurrences of a particular genotype divided by total number of sample in $\mathscr{X}$). Then the genotypic variance is simply the variance of these three values, namely V_G. The true genetic values are then calculated by

$$\mathbf{t}_g = \frac{\mathscr{X}_q \beta^*}{V_G h^2_{in} t_v}, \tag{31.4}$$

where $\mathscr{X}_q \in \{0,1,2\}^{n \times q}$ is the matrix consisting of the q columns that were chosen to be QTL, t_v is the desired output trait variance, and the division corresponds to an element-wise division of $\mathscr{X}_q \beta^* \in \mathbb{R}^n$. Here, $t_v = 40$. The vector $\mathbf{t}_g$ is then centered at zero by subtracting its mean. Finally, to calculate the actual phenotypes, $\mathbf{y}$, "environmental" noise is added to $\mathbf{t}_g$. This is done by calculating $\mathbf{y} = \mathbf{t}_g + \varepsilon$. To calculate ε, $\mathbf{e} = [e_1, \ldots, e_q]$, $e_i \sim \mathscr{N}(0, \left[t_v(1-h^2_{in})\right]^2)$ is first sampled, then

$$\varepsilon = \mathbf{e} \cdot \sqrt{\frac{(1-h^2_{in})t_v}{var(\mathbf{e})}}. \tag{31.5}$$

When estimated, the heritability of the simulated trait will be approximately equal to the desired heritability, h^2_{in}. See Figure 31.1 for a Manhattan plot describing the results of the simulation using the genotypes of 7539 sheep with 48,588 SNPs.

31.4.2 Splitting

Data splitting is an important part of wrapper based feature selection as it can control some overfitting a method displays. Here, the data is split into three groups. First, a testing set is removed from the data that will not be used in any way during the search. This will serve as external validation for the DE search process to report results. Then, with the remaining data a few choices can be made. The simplest method uses no cross-validation and splits the data again into training and valida-

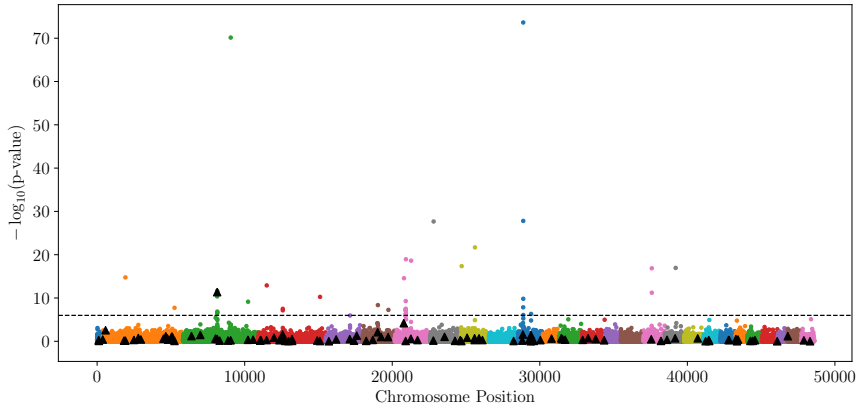


Fig. 31.1: A Manhattan plot describing the results of a GWAS on simulated data. Higher values indicate SNPs with more significant effects. Black triangles mark the randomly distributed true QTL in the simulation. The dotted line shows the Bonferroni corrected significance threshold $0.05/48588$.

tion. The training set is then used to train the BLUP model in the fitness evaluation and the validation set is used to obtain the prediction accuracy. However, since the DE search assigned fitness based on the same validation set for the entire experiment, it is likely that the search will overfit to the validation set. In practice, we used a 64%/16%/20% train/validation/test split.

To combat this, three cross-validation schemes are proposed: (1) *intergenerational* cross-validation which does k-fold cross-validation over the course of the search, changing the validation set at each generation. More concretely, at generation g_i, validation set $k \bmod g_i$ will be used to calculate prediction accuracy, and the data is used as training. (2) Intragenerational cross-validation performs k-fold cross-validation at every fitness evaluation. This method increases the computation time required by a factor of k. But, intuitively, may provide a more reliable fitness value. (3) Monte Carlo cross-validation uniformly samples a random subset of the data to be used as validation. The intuition behind this method is to drive the search through obtaining solutions that better generalize across the entire validation set. For the k-fold methods used in the following experiments, $k = 5$.

31.5 Results

The evolutionary parameters used for the results presented below are presented in Table 31.1. A fixed subset size of 1000 was used in these experiments, though more sophisticated methods can be used to choose (or even search) this value. Differential

Table 31.1: A table of evolutionary parameters used for the DE search.

Parameter (symbol)	Value
Generations (g)	5000
Population Size (N_p)	50
Crossover Rate (C_r)	0.8
Mutation Factor (F)	0.5
Replicates	10
Subset Size	1000

evolution without any additional features to improve performance will be referred to as *vanilla* DE where there is ambiguity.

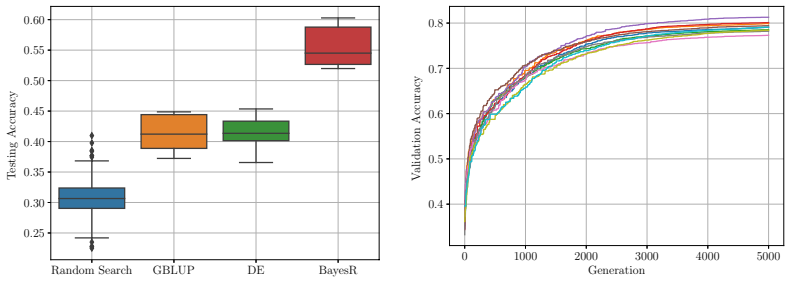
31.5.1 Baseline

As a baseline, vanilla DE will be compared to two common genomic prediction methods using the entire genome: (1) GBLUP; a functionally equivalent method to SNPBLUP. (2) BayesR; this is considered the state-of-the-art method for these exeperiments. The BayesR experiments use 50,000 iterations with a burn-in of 20,000. In addition to these methods, random search is also presented using 500 uniformly sampled subsets of size 1000 that are evaluated with SNPBLUP.

See Figure 31.2 for the results of this baseline study. It is clear that DE alone on this problem was not enough to be competitive with the state of the art method. However, there is something to be said for the comparison against GBLUP. GBLUP is essentially the fitness function for DE, which shows that the same accuracy was obtained using 50$\times$ less markers. As expected, random search provides much worse solutions than any other baseline methods, showing that DE is accomplishing something. Now that this baseline has been established, the goal is to reduce the gap between DE and BayesR.

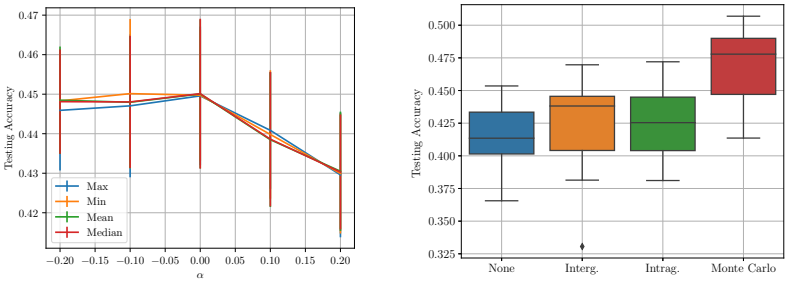
31.5.2 Controlling Overfitting

As evidence of overfitting, consider Figure 31.2(b), the fitness convergence plot for ten replicates of vanilla DE. As was discussed in Section 31.3.5, there is a theoretical maximum on performance for genomic prediction tasks. The convergence plot shows that the validation accuracy in the baseline experiment exceeded $h \approx 0.63$. This suggests that the feature subsets selected after this point were overfit to the fixed validation set—which may have led to the underperforming testing results in



(a) (b)

Fig. 31.2: (a) Boxplots comparing the vanilla DE experiment to the various baseline methods. The random search results were obtained by evaluating 500 uniformly sampled feature subsets. (b) Convergence plot for the maximum fitness at each generation for all 10 replicates in the vanilla DE experiment.



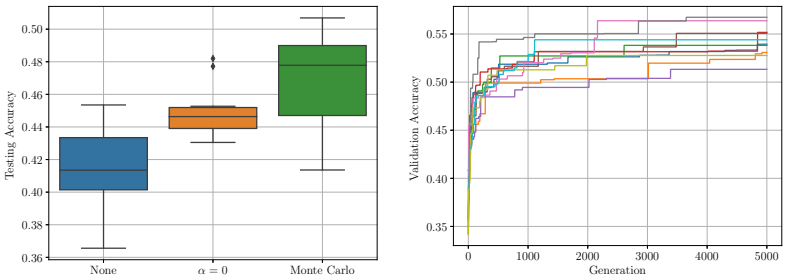
(a) (b)

Fig. 31.3: (a) Plots of testing accuracy for each statistic used with a given α in the heritability threshold formula $h(1+\alpha)$. Error bars show standard deviation. (b) Boxplots comparing the different cross-validation strategies.

Figure 31.2(a). This is a well known problem in wrapper method feature selection [47].

To remedy this, two preventative measures are proposed. First, the heritability thresholding discussed in Section 31.3.5 will be carried out for varying values of α in the formula $h(1+\alpha)$. More concretely, when some statistic of the population reaches $h(1+\alpha)$ the search will be stopped. Second, the cross-validation schemes discussed in Section 31.4.2 will be carried out as well.

The results of the thresholding experiments are presented in Figure 31.3(a). The plot shows the thresholding experiment for $\alpha \in \{-0.2, -0.1, 0, 0.1, 0.2\}$. The statistics of the population used to compare to $h(1+\alpha)$ were the minimum, maximum, mean, and median validation accuracy. It is clear that there was no significant difference between any one statistic used. However, nonnegative α values showed a



(a) (b)

Fig. 31.4: (a) Boxplots comparing the DE with no overfitting control, the $\alpha = 0$ using minimum fitness heritability threshold experiment, and DE with Monte Carlo cross-validation. (b) The convergence plot of the 10 DE with Monte Carlo cross-validation experiments.

decline in testing accuracy as α increased. This means that as validation accuracy increased past the value of h, testing accuracy decreased. There was no real trend in negative values of α.

Figure 31.3(b) shows the results of the cross-validation experiments. There was no significant difference between intergenerational and intragenerational cross-validation. However, Monte Carlo showed a significant improvement over both strategies. Some indication as to why this may have occurred is shown in Figure 31.4(b). The convergence graph for the Monte Carlo DE search shows it never reaching $h = 0.63$. Figure 31.4(a) shows the comparison of this method to the thresholding method with $\alpha = 0$. It is clear that some method to control overfitting increases testing performance over vanilla DE, however, Monte Carlo search likely has an advantage since it controls overfitting while continuing to search.

31.5.3 Improving Performance

The two methods proposed to improve performance are seeding the initial population (see Section 31.3.4) and self-adaptive DE (see Section 31.3.3). The first is intended to start the search out at a good performance using the results of a GWAS. The self-adaptive method is intended to simply boost performance through better convergence properties and to askewing the need for a stringent parameter sweep on C_r and F. In addition, these methods were combined and tested for efficacy.

Figure 31.5 shows the results of the combination experiments. Using seeding does not provide a significant boost in performance compared with vanilla DE. Intuitively, using seeding without an overfitting control mechanism simply leads to a search that converges quickly to an overfit solution. Hence, the results shown in Figure 31.5 do not show a significant difference from the unseeded experiment.

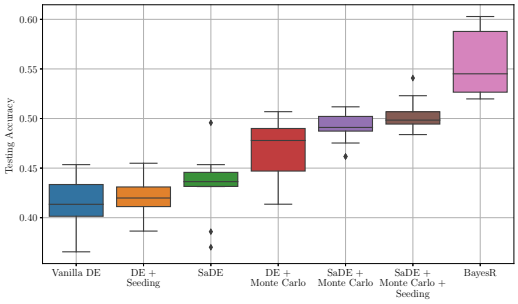


Fig. 31.5: Boxplots comparing the combination experiments and the BayesR results.

Similarly, the self-adaptive method shows a nominal improvement over vanilla DE. Once Monte Carlo cross-validation is applied to the more sophisticated self-adaptive method, the variance displayed in the SaDE experiments with no cross-validation was greatly reduced along with increasing over performance. When the self-adaptive method was combined with seeding, a slight—but not significant—increase in performance was observed. In all cases, DE underperformed when compared to the state-of-the-art method, BayesR.

31.5.4 Validation

The results presented above were a hand tuning of the DE system by adding in many components intended to increase performance. To ensure our resulting "best" configuration was not only well suited to the dataset used in the tuning experiments, a new phenotype is considered.

For this validation study, a real phenotype is used with the same sheep genotypes. In other words, our genotype is still 7,539 rows with 48,588. The heritability of the trait is estimated to be $h^2 \approx 0.16$. This value was obtained using the restricted maximum likelihood approach implemented in the NAM R package [55]. Therefore, the theoretical bound on prediction accuracy is $h \approx 0.4$.

Similarly to the baseline study, we compared DE against random search, GBLUP, and BayesR. To verify that the SaDE method with seeding and Monte Carlo does perform well, we also included it in the baseline. The results of this study are presented in Figure 31.6. The maximum obtained accuracy by the SaDE search method was 0.3993, which is on par with the estimated theoretical maximum of 0.4. However, this does not suggest DE is always better than the common methods in a non-simulated environment. As was pointed out previously, the trait being predicted is not very heritable. Because of this, it is likely that the performance of BayesR suffered greatly in comparison to the simulated environment where the effects of the QTL—and markers in linkage disequilibrium with the QTL—were large and well

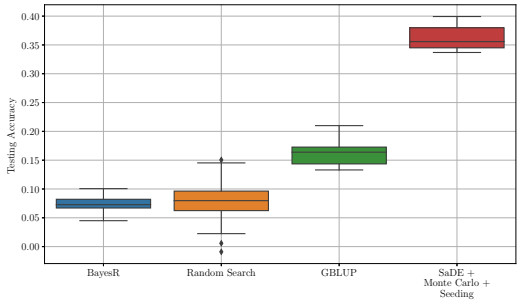


Fig. 31.6: Boxplots comparing the baseline methods to the best obtained DE method. Random search was again done by evaluating 500 uniformly sampled feature subsets.

defined. That said, these results do suggest that DE based feature selection may be better at identifying marker subsets that better capture relationships between animals, while removing noisy markers.

31.6 Discussion

With 1,000 SNP, feature selection with DE performed on par with BayesR and outperformed RRBLUP, even though the latter is mostly due to the structure of the simulation – with real data, the differences are generally negligible. In practice, at this point, we expect the three methods to be largely comparable with each other. But it is noteworthy that by using the DE, the same results are achievable with 1,000 SNP instead of almost 50,000. The DE captured a reasonable proportion of the *real* underlying causal variants and other features in the data that approximated the genetic architecture. This does bring us closer to prediction models based on real causative variants which has several advantages in relation to whole-genome models: 1) there is an immediate benefit for industry to have smaller panels since the production costs are lower and it will enable wider adoption of the technology; 2) whole-genome methods rely on relationships between individuals, the accuracy of prediction in distantly related ones is very low – functional panels can be expected to hold accuracy irrespective of the genetic distances between the discovery and validation populations; 3) can provide new biological insights and novel candidate targets for clinical intervention.

However, performance is still lacking. We saw good results with 1,000 SNP but of course, for this particular data set the objective was to evolve a panel of high accuracy with only 100 SNP. Even the panel with 1,000 SNP only included 15 QTL, with a sizable proportion of the accuracy still coming from the DE using the SNP to optimize the relationship structure between individuals in a manner that maximized the accuracy. For comparison purposes it is interesting to note that simply using the

top 100 SNP from the GWAS does a better job than DE, RRBLUP or BayesR. This is, by design, just an artifact of the simulation since many QTL have very high LOD scores (Figure 31.1) which are much higher than usually observed with complex polygenic traits; but it does suggest that there is scope to use GWAS results to *seed* the DE runs in the future.

We believe that the underperfomance of the DE with 100 SNP is largely attributable to overfitting of the data. During the search, there is only a finite amount of data to evaluate and assign fitness with and the search is guided by the performance on the validation set. Therefore, it is more likely to select SNPs that perform well on the validation data set alone, leading to overfitting. More evidence of this comes from observations on the heritability of our trait. Because $h^2 = 0.3855$, we know that about 39% of the variation in the phenotype is accounted for by the effects from the QTL in the genotypes. We can expect our best possible accuracy to be $h^2/\sqrt{h^2}$. Figure 31.7 shows the convergence of the 1,000 feature subset DE experiment. This shows our search obtaining values much higher than this expected maximum accuracy of 0.6208, which of course should not be possible. Hence, our search overfits to our validation set, which is a well known problem of wrapper methods in feature selection [47].

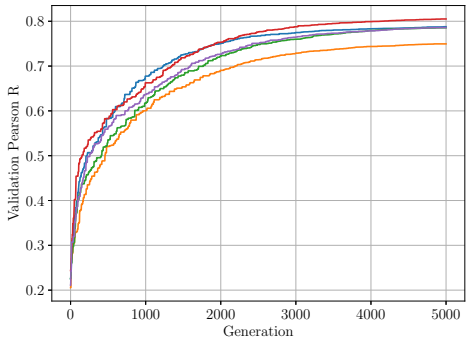


Fig. 31.7: Convergence plot for the 1000 feature subset DE experiment. The separate lines show the five replicates.

The smaller subset experiments likely converge into local optima quickly and their searches stall before finding good solutions. This may be due to the comparative dimensionalities of the encoding and the solution space. More precisely, we care only about 10 or 100 entries out of a vector in $\mathbb{R}^{48,588}$, which may be too large of a discrepancy to find any promising solutions. This could cause the search to converge to the first promising candidate solutions and not break out of the local optimum – both subsets of size 10 and 100 rapidly converged onto the 4 largest QTL. While for this work we wanted to evaluate the algorithm with hardset feature numbers, in real world scenarios it is better to co-evolve the number of selected features as an additional parameter in the algorithm.

31.7 Future Work

It is clear that applying domain knowledge is a promising avenue for DE search in this application. Evolutionary computation approaches often can easily accommodate prior knowledge about a problem to enhance their performance. A further opportunity for applying domain knowledge is to consider epistasis. Our method, in its current form, would not effectively find feature subsets with epistatic effects since they are not explicitly modelled by RRBLUP; even though in practice, they are implicitly captured to some extent through the genetic relationships in closely related populations. But more broadly, epistatic effects are non-linear and therefore cannot be captured with ridge regression. However, with minimal effort, a model that can capture non-linear effects—e.g., a small neural network—can be used in the DE fitness function to find feature subsets corresponding to markers that have epistatic interactions with each other.

We believe that alternative AI approaches like DE will become necessary when applying genomic prediction to full sequence data. Genome-wide association studies are well known to not be able to identify many variants that all have small effects [30] and the multiple testing problem with millions of variants will further confound the issue. For genomic prediction, there is very little to gain from sequence data with RRBLUP (or the equivalent GBLUP) as the changes to the genomic relationship matrix are minimal, and consequently the predictions will essentially be the same even if millions of additional SNP are included in the model. Bayesian approaches should be better able to discern effects and it is expected that these methods should have higher accuracy, in the same way as BayesR did with our simulation – but they are currently computationally intractable at the sequence level.

31.8 Conclusion

We have presented a competitive feature selection algorithm for genomic prediction problems. Currently, DE performs competitively with genome-wide methods using a fraction of the number of SNP and seems a promising alternative to current prediction methods. Feature selection with DE, or any other method, has the advantage of giving an interpretable result. With the rise of complex non-linear approaches like deep convolutional and recurrent neural networks becoming popular in bioinformatics and biomedicine, a precise and readable result could lead to a more complete understanding of the complex genotype-phenotype mapping. Differential evolution presents a non-statistical alternative to current state of the art methods with potential to easily apply domain knowledge to this difficult problem.

Acknowledgements We dedicate this chapter to BEACON's director Erik Goodman – advisor, colleague, mentor and friend. This work was supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454 (BEACON 2012); by the Next-Generation BioGreen 21

Program (Project No. PJ01322204), Rural Development Administration, Republic of Korea and by the National Institute of Food and Agriculture (AFRI Project No. 2019-67015-29323).

References

1. Abraham, G., Tye-Din, J.A., Bhalala, O.G., Kowalczyk, A., Zobel, J., Inouye, M.: *Accurate and robust genomic prediction of celiac disease using statistical learning*. PLoS Genetics **10**, e1004137 (2014)
2. Al-Mamum, H.A., Kwan, P., Clark, S., Lee, S.H., Song, K.D., Lee, S.H., Gondro, C.: *Genomic best linear unbiased prediction using differential evolution*. In: Proceedings of the AAABG 21st Conference, pp. 145–148 (2015)
3. Altidor, W., Khoshgoftaar, T.M., Van Hulse, J.: *Robustness of filter-based feature ranking: A case study.* In: Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society (FLAIRS) Conference, pp. 453–458 (2011)
4. Banzhaf, W., Nordin, P., Keller, R., Francone, F.: *Genetic Programming - An Introduction*. Morgan Morgan Kaufmann Publishers Inc. (1998)
5. Bean, J.C.: *Genetic algorithms and random keys for sequencing and optimization*. ORSA Journal on Computing **6**(2), 154–160 (1994)
6. Bhattacharyya, S., Sengupta, A., Chakraborti, T., Konar, A., Tibarewala, D.N.: *Automatic feature selection of motor imagery EEG signals using differential evolution and learning automata*. Medical & Biological Engineering & Computing **52**(2), 131–139 (2014)
7. Biesiada, J., Duch, W.: *A Kolmogorov-Smirnov Correlation-Based Filter for Microarray Data*. In: M. Ishikawa, K. Doya, H. Miyamoto, T. Yamakawa (eds.) Neural Information Processing, pp. 285–294. Springer (2008)
8. Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J.M., Herrera, F.: *A review of microarray datasets and applied feature selection methods*. Information Sciences **282**, 111–135 (2014)
9. Clark, S.A., Hickey, J.M., van der Werf, J.H.: *Different models of genetic variation and their effect on genomic evaluation*. Genetics Selection Evolution **43**(1), 18 (2011)
10. Das, S., Suganthan, P.: *Differential evolution: A survey of the state-of-the-art.* IEEE Transactions on Evolutionary Computation **15**, 4–31 (2011)
11. De Los Campos, G., Naya, H., Gianola, D., Crossa, J., Legarra, A., Manfredi, E., Weigel, K., Cotes, J.M.: *Predicting quantitative traits with regression models for dense molecular markers and pedigree*. Genetics **182**(1), 375–385 (2009)
12. Ding, C., Peng, H.: *Minimum redundancy feature selection from microarray gene expression data*. Journal of Bioinformatics and Computational Biology **3**(02), 185–205 (2005)
13. Dorigo, M., Maniezzo, V., Colorni, A.: *Ant system: Optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **26**(1), 29–41 (1996)
14. Eiben, A., Smith, J.E.: *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg (2003)
15. Erbe, M., Hayes, B.J., Matukumalli, L.K., Goswami, S., Bowman, P.J., Reich, C.M., Mason, B.A., Goddard, M.E.: *Improving accuracy of genomic predictions within and between dairy cattle breeds with imputed high-density single nucleotide polymorphism panels*. Journal of Dairy Science **95**(7), 4114–4129 (2011)
16. Esquivelzeta-Rabell, C., Al-Mamum, H.A., Lee, S.H., Song, K.D., Gondro, C.: *Evolving to The Best SNP panel for Hanwoo Breed Proportion Estimates*. In: Proceedings of the AAABG 21st Conference, pp. 473–476 (2015)
17. Firpi, H.A., Goodman, E.: *Swarmed feature selection*. In: 33rd Applied Imagery Pattern Recognition Workshop (AIPR'04), pp. 112–118 (2004)

18. Forneris, N.S., Legarra, A., Vitezica, Z.G., Tsuruta, S., Aguilar, I., Misztal, I., Cantet, R.J.C.: *Quality control of genotypes using heritability estimates of gene content at the marker*. Genetics **199**(3), 675–681 (2015)
19. Goddard, M.: *Genomic selection: Prediction of accuracy and maximisation of long term response*. Genetica **136**(2), 245–257 (2009)
20. Goddard, M., Hayes, B., Meuwissen, T.: *Using the genomic relationship matrix to predict the accuracy of genomic selection*. Journal of Animal Breeding and Genetics **128**(6), 409–421 (2011)
21. Goddard, M.E., Hayes, B.J.: *Genomic selection*. Journal of Animal Breeding and Genetics **124**(6), 323–330 (2007)
22. Goldberg, D.: *Genetic Algorithms in Search, Optmization and Machine Learning*. Addison Wesley (1989)
23. Gondro, C.: *Primer to Analysis of Genomic Data Using R*. Springer International Publishing (2015)
24. Habier, D., Fernando, R.L., Dekkers, J.C.M.: *The impact of genetic relationship information on genome-assisted breeding values*. Genetics **177**(4), 2389–2397 (2007)
25. Habier, D., Fernando, R.L., Garrick, D.J.: *Genomic BLUP Decoded: A Look into the Black Box of Genomic Prediction*. Genetics **194**(3), 597–607 (2013)
26. Habier, D., Fernando, R.L., Kizilkaya, K., Garrick, D.J.: *Extension of the Bayesian alphabet for genomic selection*. BMC Bioinformatics **12**(1), 186 (2011)
27. Hayes, B., Bowman, P., Chamberlain, A., Goddard, M.: *Invited review: Genomic selection in dairy cattle: Progress and challenges*. Journal of Dairy Science **92**(2), 433 – 443 (2009)
28. Hayes, B.J., Pryce, J., Chamberlain, A.J., Bowman, P.J., Goddard, M.E.: *Genetic architecture of complex traits and accuracy of genomic prediction: Coat colour, milk-fat percentage, and type in Holstein cattle as contrasting model traits*. PLoS Genetics **6**, e1001,139 (2010)
29. van Heel, D.A., Franke, L., Hunt, K.A., Gwilliam, R., Zhernakova, A., Inouye, M., Wapenaar, M.C., Barnardo, M.C.N.M., Bethel, G., Holmes, G.K.T., Feighery, C., Jewell, D., Kelleher, D., Kumar, P., Travis, S., Walters, J.R., Sanders, D.S., Howdle, P., Swift, J., Playford, R.J., McLaren, W.M., Mearin, M.L., Mulder, C.J., McManus, R., McGinnis, R., Cardon, L.R., Deloukas, P., Wijmenga, C.: *A genome-wide association study for celiac disease identifies risk variants in the region harboring IL2 and IL21*. Nature Genetics **39**, 827–829 (2007)
30. Hirschhorn, J.N., Daly, M.J.: *Genome-wide association studies for common diseases and complex traits*. Nature Reviews Genetics **6**(2), 95–108 (2005)
31. Holland, J.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, Ann Arbor (1975)
32. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: *An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **42**(2), 482–500 (2012)
33. Khushaba, R.N., Al-Ani, A., Al-Jumaily, A.: *Differential evolution based feature subset selection*. In: 2008 19th International Conference on Pattern Recognition, pp. 1–4 (2008)
34. Khushaba, R.N., Al-Ani, A., AlSukker, A., Al-Jumaily, A.: *A combined ant colony and differential evolution feature selection algorithm*. In: M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, A.F.T. Winfield (eds.) Ant Colony Optimization and Swarm Intelligence, pp. 1–12. Springer (2008)
35. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA (1992)
36. Kwong, Q.B., Ong, A.L., Teh, C.K., Chew, F.T., Tammi, M., Mayes, S., Kulaveerasingam, H., Yeoh, S.H., Harikrishna, J.A., Appleton, D.R.: *Genomic Selection in Commercial Perennial Crops: Applicability and Improvement in Oil Palm (Elaeis guineensis Jacq.)*. Scientific Reports **7**, 2872 (2017)
37. Luque-Baena, R., Urda, D., Claros, M.G., Franco, L., Jerez, J.: *Robust gene signatures from microarray data using genetic algorithms enriched with biological pathway keywords*. Journal of Biomedical Informatics **49**, 32 – 44 (2014)

38. Luque-Baena, R.M., Urda, D., Subirats, J.L., Franco, L., Jerez, J.M.: *Application of genetic algorithms and constructive neural networks for the analysis of microarray cancer data*. Theor Biol Med Model **11**(Suppl 1), S7–S7 (2014)
39. Meuwissen, T.H.E., Hayes, B.J., Goddard, M.E.: *Prediction of total genetic value using genome-wide dense marker maps*. Genetics **157**(4), 1819–1829 (2001)
40. Moghaddar, N., Swan, A.A., Van Der Werf, J.H.J.: *Comparing genomic prediction accuracy from purebred, crossbred and combined purebred and crossbred reference populations in sheep*. Genetics Selection Evolution **46**, 58 (2014)
41. Nearchou, A.C., Omirou, S.L.: *Differential evolution for sequencing and scheduling optimization*. Journal of Heuristics **12**(6), 395–411 (2006)
42. Patterson, H.D., Thompson, R.: *Recovery of inter-block information when block sizes are unequal*. Biometrika **58**(3), 545–554 (1971)
43. Peng, H., Long, F., Ding, C.: *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy*. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8), 1226–1238 (2005)
44. Qin, A.K., Suganthan, P.N.: *Self-adaptive differential evolution algorithm for numerical optimization*. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005)
45. Raymer, M.L., Punch, W.F., Goodman, E.D., Kuhn, L.A., Jain, A.K.: *Dimensionality reduction using genetic algorithms*. IEEE Transactions on Evolutionary Computation **4**(2), 164–171 (2000)
46. Rechenberg, I.: *Evolutionsstrategie*. Holzmann-Froboog, Stuttgart (1975)
47. Saeys, Y., Inza, I.n., Larrañaga, P.: *A review of feature selection techniques in bioinformatics*. Bioinformatics **23**(19), 2507–2517 (2007)
48. Schwefel, H.P.: *Evolution and Optimum Seeking*. Wiley (1995)
49. Storn, R., Price, K.: *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization **11**(4), 341–359 (1997)
50. Su, G., Brndum, R., Ma, P., Guldbrandtsen, B., Aamand, G., Lund, M.: *Comparison of genomic predictions using medium-density (54,000) and high-density (777,000) single nucleotide polymorphism marker panels in Nordic Holstein and Red Dairy Cattle populations*. Journal of Dairy Science **95**(8), 4657 – 4665 (2012)
51. Van Raden, P.M.: *Efficient methods to compute genomic predictions*. Journal of Dairy Science **91**(11), 4414–4423 (2008)
52. Visscher, P.M., Wray, N.R., Zhang, Q., Sklar, P., McCarthy, M.I., Brown, M.A., Yang, J.: *10 Years of GWAS Discovery: Biology, Function, and Translation*. The American Journal of Human Genetics **101**(1), 5 – 22 (2017)
53. Whittaker, J.C., Thompson, R., Denham, M.C.: *Marker-assisted selection using ridge regression*. Genetical Research **75**(2), 249–252 (2000)
54. Wientjes, Y.C.J., Veerkamp, R.F., Calus, M.P.L.: *The effect of linkage disequilibrium and family relationships on the reliability of genomic prediction*. Genetics **193**(2), 621–631 (2013)
55. Xavier, A., Xu, S., Muir, W., Rainey, K.: *NAM: Association studies in multiple populations*. Bioinformatics **31**(23), 3862–3864 (2015)
56. Zapata-Valenzuela, J., Whetten, R.W., Neale, D.B., McKeand, S.E., Isik, F.: *Genomic Estimated Breeding Values Using Genomic Relationship Matrices in a Cloned Population of Loblolly Pine*. G3: Genes, Genomes, Genetics **3**, 906–916 (2013)