Machine Learning and Computational Intelligence for High-Order Epistasis Detection

Marco Araújo Stobberup Dias da Graça
marco.graca@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2019

Abstract

Genome-Wide Association Studies (GWAS) aim to discover how genetic markers like Single Nucleotide Polymorphisms (SNP) mutually interact with each other, enabling the manifestation of diseases or traits. Full exhaustive analysis is prohibitive due to the exponentially increasing, with the numbers of markers assumed to be involved and SNPs present in the human genome, of interactions to test. Common approaches use machine learning techniques to rank SNPs according to their estimated predictive power or create clusters for SNPs assumed to produce highly scoring interactions between them. The proposed approach uses a genetic algorithm to search the parameter space of a gradient boosting machine in order to find an optimal configuration. This configuration will produce a model in which the most prominent features will coincide with the SNPs involved in interactions of highly predictive power for determining wether a patient will manifest a certain trait or not, prompting an exhaustive search on that subspace to find best scoring interactions. Optimal solutions for interactions of up to 5 SNPs where found or closely approached with this approach on toy datasets where interactions of superior order where hidden with controllable prediction power for the studied trait. The algorithm is an inference tool for GWAS that works outside of common interaction orders studied, being able to find promising higher-order interactions for further biological study without an exponential increase in time.
Keywords: Genetic-Wise Associaton Study; Higher-Order Epistasis; Genetic Algorithm; Exhaustive Search; Interaction Order.

1. Introduction

The methods for acquiring comprehensive genetic polymorphism datasets, representing occurrences of alleles of genes within a population, progressed considerably. This has facilitated the discovery of associations between a number of diseases and certain genetic markers, such as Single Nucleotide Polymorphisms (SNPs). Comprehensive analysis in datasets containing the expression of numerous genetic markers in a genome-wide association study (GWAS) is now feasible and not as time-consuming, even if considering a large population of individuals. The same cannot be said for rigorous studies that aim to detect if the phenotypic effect of a certain marker is influenced by the expression of one or more other markers (epistasis), which, despite less common than studies that focus on the phenotypic effects of isolated markers, have proven to be of great interest for the biochemistry and genetic science communities [4][22]. These studies uncovered significant findings on the heritability and gene regulatory network behind the manifestation of several diseases, such as Crohn's disease[10], Alzeihmer[20] and obesity[6].

Performing an exhaustive search on a dataset for epistatic interactions that might result in the phenotypic trait being studied is still burdened on interactions of higher order by an exponential increase in time complexity, due to the combinatorial nature of the posed problem. Nevertheless, it is still a relevant procedure that could lead to a better understanding of the genetic architecture of complex diseases.

New state-of-the art approaches take advantage of machine learning innovations, genetic algorithms and artificial intelligence to produce alternatives to an exhaustive search that can still uncover significant epistatic interactions while drastically reducing time complexity. However, these approaches are still mainly focused on pairwise interactions.

With the use of machine learning methods to enhance searches and the use of heuristics for Artificial Intelligence (AI) powered searches, new challenges arise, possibly halting the search on sub-optimal interactions, creating models on the SNPs interaction power that overfit the data and produce erroneous results and, for low complexity models, the costly need of post-processing stages for the removal of

false positives.
We introduce a method that aims to synergize the combined efforts of a genetic algorithm for tuning search parameters, a machine learning approach as a filtering stage and an exhaustive search on a given subspace of the data as a way to efficiently detect significant higher-order epistatic interactions on feasible runtime.

1.1. Motivation

While most works on genetic interactions focus on relatively simple cases involving only two loci, i.e., fixed positions of genes or genetic markers on a chromosome, high-order genetic interactions involving three or more loci also occur and can have major phenotypic effects[8]. Works that aim to map changes in protein functions or noticeable traits, such as disease manifestation, would benefit from a wider availability of tools, namely ones that harness the potential of the latest technological advancements.
Current exhaustive search algorithms suffer from an exponential increase in time and space complexity regarding the order of the epistatic interaction to detect. Datasets often represent hundreds of thousands or even millions of SNPs. In such cases, the only feasible option to produce fruitful results in reasonable time-to-solution is to perform heuristic-guided searches on reduced search spaces. Machine Learning (ML) advancements and AI based algorithms provide such heuristics, that while performing an incomplete search, can still detect pertinent higher order interactions.
Most heuristically guided algorithms that work on higher order interactions build on promising lower order interactions in an effort to use a subset of the SNPs involved in the lower interaction to reduce the search space of the higher order[24] [23]. This approach assumes that optimal solutions can be found by recursively improving results found on lower orders and thus does not consider other forms of epistasis where strong epistatic interactions are present between SNPs who possess unnoticeable interaction power on lower orders.
We intend to avoid compromising the improved time-to-solution typically associated with doing an incomplete, but time efficient, search on a dataset for interactions of a user-defined order by overfitting to local optimal solutions of specific orders. By enabling a search for several solutions of different orders, where the information gathered for each solution is the configuration of the filtering stage that lead to it, not the SNPs, we aim to provide our framework with tools that avoid trapping on local optima and a continuous chance of improvement on the solutions.

2. Background in Epistasis and Machine Learning

The genetic material of an organism, its genome, is comprised of nucleotides, the building blocks of DNA (and RNA) molecules, which carry crucial genetic instructions for the functioning of all forms of life. A DNA sequence is built upon an alphabet of four different units, the different nitrogenous bases a nucleotype can possess (A, C, G, T). To a variation among individuals of a single nucleotide in a certain position, when prevalent in at least 1% of a population, we call Single Nucleotide Polymorphism (SNP) [2]. A sequence of nucleotides that code a molecule that performs a function is called a gene. If an SNP occurs within a given gene, it is said to have more than one allele or variant form. A presence of different observable traits on a population can be a result of the individuals in said population having different alleles of the same gene. The effects of different alleles at one gene can be dependent on the allele of another gene or genes, denoting an epistatic relationship or interaction between the SNPs that distinguish said alleles.
SNPs are one of the most popular classes of genetic markers used in GWAS for the discovery of genotype-phenotype associations, where a combination of alleles of interest an individual possesses (its genotype) is assumed to be the cause behind how said individual differs from others. In organisms with two matching sets of chromosomes (diploid), one allele is inherited from the male parent and other from the female one, both in the same fixed position of their respective chromosome - its locus. If both alleles of a diploid organism are the same, the organism is homozygous at that locus. If they are different, the organism is heterozygous at that locus. For the same gene, the effect on phenotype of one allele, the dominant allele, masks the contribution of a second allele, the recessive, at the same locus. The studied genotypes of a SNP are usually coded as $\{0, 1, 2\}$ corresponding to **homozygous major** genotype (e.g. AA, BB), where the individual posesses two copies of the allele that codes for the dominant trait, **heterozygous** genotype (e.g. aA, Aa, bB, Bb), and **homozygous minor** genotype (e.g. aa, bb), for individuals with two copies of the recessive allele, respectively. The label of an individual is a binary phenotype, being either 0 (control) if the individual does not manifest the trait in question or 1 (case) if otherwise.

2.1. Role in Evolution

Common complex diseases, such as Alzheimer's disease[5], breast cancer[15], or diabetes[11], are known to be influenced by more than one gene. GWAS have been successful, for some traits/phenotypes, with the identification of new disease susceptibility genes in a one by one test for

common SNPs across the human study population genome, while other diseases have been less successful through single-SNP GWAS. Single SNP variations, also called the marginal effect of said SNPs, fail to explain the heritability of a phenotype, an estimate on how, between the individuals of a population, the variation of a phenotypic trait is due to genetic SNP variation[9].

A common statistic tool used to better understand how the data is clustered and draw causality relations is a contingency table, a matrix where the dimensions correspond to variable distributions from which we wish to draw correlations. In the contingency tables of Figure 1 we include a comparative simulated example of where a statistically significant epistatic interaction takes place between two SNPs where no marginal effect was detected individually. On the two graphs on the left, where

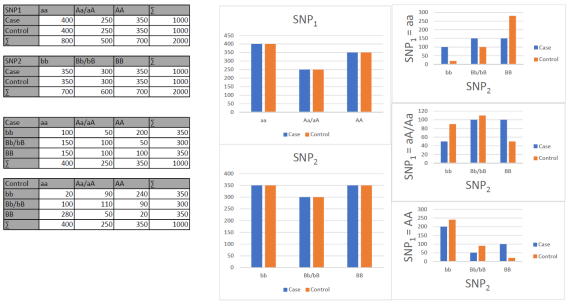

Figure 1: SNP_1 and SNP_2 Contingency Tables

only the state distribution of a single SNP is considered, the number of cases and controls is evenly split for all their possible combinations, which would suggest on a surface analysis that neither of these SNPs are relevant for the variation of the studied phenotype. Conversely, the three graphs on the right, where the phenotype variation is studied on SNP_2, taking into account the genotypic state of SNP_1 as well, shows more disparity.

2.2. Biologic and Statistical Epistasis

Biological epistasis refers to "the result of physical interactions among biomolecules within gene regulatory networks and biochemical pathways in an individual such that the effect of a gene on a phenotype is dependent on one or more other genes"[12]. The phenomenon of statistical epistasis refers instead to the discovery of relationships between multilocus genotypes, namely the frequency distribution according to the three mentioned possible states of a SNP, and phenotypic variation. Differences in biological epistasis among individuals in a population give rise to statistical epistasis.

Often, logistic regression is employed for epistatic detection in case-control GWAS as a statistical method for relating a linear combination of SNP values and the probability (p) of that combination manifesting alongside the studied trait. Given a dataset, we may fit a model where the x_1 and x_2 variables represent the input variables (SNPs) whose pairwise interaction we wish to test:

$$log\left(\frac{p_{12}}{1-p_{12}}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 \quad (1)$$

where β_0 denotes the mean term of the model which is determined by population prevalence and data size; β_0,β_1 and β_{12} are modeled as the main effects and interactive effects, respectively; p_{12} denotes the conditional probability $p(y = 1|x_1, x_2)$ and the Odds Ratio (OR) is $\frac{p}{1-p}$. Proving causation will always be a challenge due to an inability to randomly assign people to genotypes as it is possible with model organisms. However, evidence in favor of an association can be significantly strengthened through comprehensive efforts to address sources of error and bias[12], such as staging statistical tests on formed hypotheses or gathering more individuals for the genome-wide study.

Bayesian networks, such as the one represented in

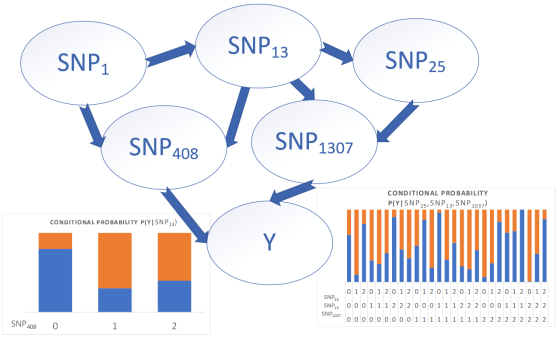

Figure 2: Bayesian Network

Figure 2, provide a compact representation of the dependencies between variables, consisting of a directed acyclic graph, the disease model, where the variables (SNPs) are represented by nodes and dependencies between them for phenotypic expression are represented by edges. In the figure, some nodes display information on how the states of their parent nodes influence their conditional probability in the annexed bar graphics. Each bar aims to show the frequency of cases, in orange, and controls, in blue, conditioned on the state of the other nodes, as described by the combination of SNP states below each bar.

2.3. Challenges

The aim of this thesis focuses on the challenges that arise with detecting epistatic interactions of higher order that are statistically relevant for the presence of the studied phenotypic traits, namely the problem that comes when trying to scale up algorithms

that detect gene-gene interactions (k = 2) to higher interactions. As an example, the time needed for calculating the mutual information on a toy dataset, in a MATLAB script, is shown in Figure 3. A more detailed explanation on this measure is provided on the section 2.4. For different values of k (number

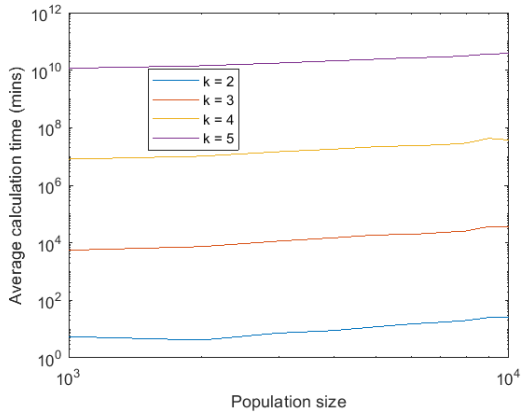

Figure 3: Estimation of Time in Minutes for the Calculation of Mutual Information

of SNPs involved in the measured interaction) the figure denotes, on a logarithmic scale, in function of the simulated number of individuals, that a linear increase in k is followed by an exponential increase in the time complexity. For this dataset, only thousands of SNPs were considered, but a typical genome differs from the reference human genome in about 4 to 5 million sites[3], setting the usual number of SNPs for such studies at higher orders of magnitude and exponentially costlier in time.

2.4. Objective Functions

For comparative purposes, algorithms rely on objective functions that score quantitatively the effect of an interaction between the SNPs in a subset given as input, by using mathematical formulas to relate the frequency of the different combinations of SNP genoptypic combinations to the conditional frequency of the manifested trait. The aim of an objective function is to give a scalar score to an association, based on the case-control distribution of the studied population across the several states of the SNPs in the considered association.

K2 Score

The K2 algorithm heuristically searches for the most probable Bayesian Network structure given a subset of SNPs and their instantiations across the dataset plus the phenotype states as classification targets. The nodes of the network will therefore be the SNPs of the subset with the trait states as leaves of the network[16].

$$f(i,\pi_i) = \sum_{j=1}^{q_i}\left(log\left[\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\right] + \sum_{k=1}^{r_i} log(a_{ijk}!)\right) \quad (2)$$

where:

π_i: Set of parent SNPs of SNP_i;

q_i: Number of all possible different genotype combinations of the parents of SNP_i in the input database (3^s where s is the number of elements on the set π_i)

r_i: Number of possible states (2 for trait nodes due to the case-control nature of the GWAS and 3 for SNP nodes);

a_{ijk}: number of instances in the dataset in which the attribute SNP_i is instantiated with its k^{th} value, and the parents of SNP_i in π_i are instantiated with the j^{th} instantiation;

N_{ij}: Number of instances in the database in which the parents of SNP_i in π_i are instantiated with the j^{th} instantiation ($\sum_{k=1}^{r_i} a_{ijk}$).

Information Theory measures

Information entropy has been defined as the average amount of information that is produced by a stochastic source of data that can be used to measure the data distribution diversity, and to measure the uncertainty of random variables[17]. Let $p(x_i)$ be the probability of the i^{th} genotype of a SNP variable (X), observed in n individuals, the entropy of X is expressed as:

$$H(X) = -\sum_{i=1}^{n} p(x_i)log(p(x_i)) \quad (3)$$

However, entropy does not assess any information on the hypotheses that several variables are either correlated or independent. For that purpose, the mutual information (MI) of two variables is more adequate, measuring the mutual dependence between the two variables (amount of "information" that one variable contains about the other) by how similar its joint distribution is to the product of the marginal distributions of both[21]. For example, let X be an SNP locus and Y a phenotype trait, the mutual information of both variables can be described as:

$$I(X;Y) = \sum_{i=0}^{2}\sum_{j=0}^{1} p(x_i,y_j)log(\frac{p(x_i,y_j)}{p(x_i)p(y_j)}) = H(X)+H(Y)-H(X,Y) \quad (4)$$

2.5. Machine Learning and Artificial Intelligence

The higher dimensionality and combinatorial nature of the posed problem imply that no complete search for higher-order ($k > 2$) epistatic interactions will be attainable on the current scale of GWAS datasets. If we consider the SNP data as input and the trait to study as the output, the objective of a GWAS can be interpreted as finding a function or mathematical model that computes the output from any given input, a supervised classification problem. ML helps performing such a task

by iteratively reducing the error between the function output and the ground truth contained in the dataset by updating the set of parameters that define the model function to build.

The choice of ML methods that trade accuracy for low complexity as pre-processing stages allows for a more diverse range of algorithms to integrate in a full method.As an example, should the model return a scalar weight for different SNPs, a further exhaustive search can be conducted on a sorted list of SNPs by their attributed weights. When exploring adding a ML approach to a stage of an interaction detection algorithm, the possibility of overfitting the resulting model must be taken into account. This occurs when the complexity of the algorithm, allied with the noise that comes with the chosen dataset makes the training stage overly sensitive to the peculiarities of the dataset. Its predictive power is then unable to be reproduced on any other input, rendering the produced model unfit.

2.6. Decision Trees, Random Forests and Gradient Boosting

Tree-based algorithms generate a directed acyclic graph (DAG) where each node represents an input variable (an SNP for GWAS data) and leaf nodes represent a target classification (phenotypic trait selection). They differ from bayesian networks as the outputted model is deterministic and not stochastic. In the Figure 4, the topmost decision is made by the SNP_2 node, the root of the tree, which means that the greatest marginal effect for the presence of the studied phenotypic trait comes from that SNP, since it generates the least amount of entropy (see section 2.3.1). A tree is learned from then on by "splitting" the dataset into subsets based on a value test for an input variable (i.e. $SNP_2 = 2$ or $SNP_1 \neq 0$) in a recursive manner until no further discriminating value can be added to the prediction, represented by the branch currently examined.

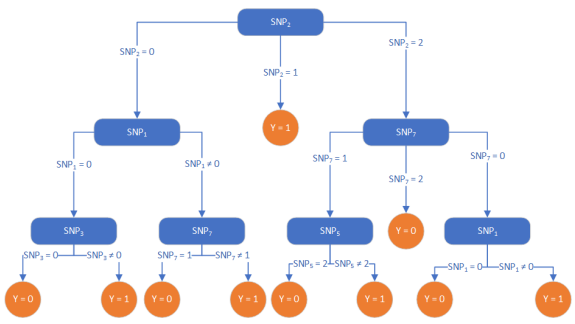

Figure 4: Example of decision tree grown from a GWAS dataset

Random forests help avoiding the bias of selecting only one node as the root upon which all future splits are conditioned by growing an ensemble of trees and choosing the most prevalent trait in the output of all trees. Furthermore, each tree is trained on a new dataset, sampled uniformly and with replacement, from the original dataset (bootstrap sample) and at each node, the best split is chosen from a randomly selected subset of the SNP variables (feature bagging). By combining the prediction of a diverse set of trees, bagging utilizes the fact that classification trees are unstable but on average produce the right prediction.

Another method that takes advantage of an ensemble of weak predictors that combined make a stronger predictor is a Gradient Boosting Machine (GBM), which trades the bagging of features common to random forests algorithms for "boosting". In other words, the predictors are made sequentially, not independently, and every new predictor learns from the residual error of the previous one, in an effort to reduce variance and individual variable bias and possibly hasten learning, but heightening overfitting risk. Here the weak predictors can be chosen from a wide range of classifiers or regressors, usually decision trees.

2.7. Genetic Programming Algorithms

Algorithms for SNP interaction detection have also started to take advantage of evolutionary programming heuristics, like swarm intelligence [7][19][18] or logic regression using genetic programming[13]. On a genetic algorithm (GA), a set of individuals called population undergoes adaptions. From there, a selection process based on fitness leads to a new generation of individuals. The common genetic programming algorithm behavior is to:

1. Create an initial random population;
2. Perform the following steps on the current generation:
   (a) Select individuals in the population based on a selection scheme;
   (b) Adapt the selected individuals;
   (c) Evaluate the fitness value of the adapted individuals.;
   (d) Select individuals for the next generation according to a selection scheme;
3. If the termination criterion is fulfilled, then output the final population. Otherwise, set the next generation as current and go to step 2.

The set of candidate individuals for a new generation is constructed by selecting one of three adaptation operations for each candidate individual:

Reproduction: Copying individuals from the current generation.

Crossover: Combining monomials of two randomly chosen individuals to create a new individual.

Mutation: Applying a random change to an individual, such as inserting a new literal, deleting

a literal, replacing a literal by a new literal, inserting a new literal as a new monomial or deleting a monomial.

3. Framework for Higher-Order Epistasis Detection

We propose an approach (see fig. 5) that mitigates the time complexity of an exhaustive search on a given database relying on a GBM (1) whose parameters are tuned by a GA (2), in an effort to avoid being trapped in a local sub-optimal search.
The GA randomly initializes a set of parameters for the later stages as an individual. These receive feedback of their performances in the form of the fitness of the resulting GBM model and the interaction relevance of the exhaustive search provided solution. Based on said information, the search will create new offspring individuals via genetic operators (i.e., mutation, crossover).
The GBM (1) we implement consists of an ensemble of decision trees with customizable number, depth and other learning parameters. A learning and testing set split of the data is employed as caution against overfitting. The outputted models then serve as a filtering stage, ranking SNPs according to their computed conditional variable importance. The chosen SNPs for further exhaustive search on a reduced space follow a customizable parameter as well, a defined number of SNPs to take from the top of the sorted list: top_feats.

For the exhaustive search on the subset resulting from the GBM stage, an objective function is then employed as an indicator to compare each score on the relevance and power of the proposed interactions and biological significance of said solutions.

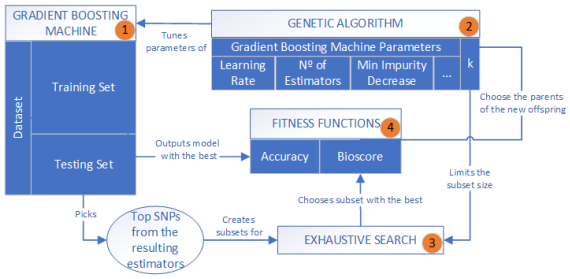

Figure 5: Architecture of the Proposed Algorithm

3.1. Initial XGBoost Parameter Configuration

XGBoost, as we intend to incorporate in our algorithm, using the DART (Dropout meets Multiple Addictive Regression Trees[14]) booster has several parameters we must take in consideration to decide on the best starting configuration, what parameters[1] to tune using the genetic algorithm and which boundaries to impose on those:

subsample Fraction of the patients to consider at each iteration.

colsample_bytree/level/node Fraction of the SNPs to consider at each iteration. The sampling can occur at every different tree being grown (*tree*), at each depth level of said tree (*level*) or at the growth of every single tree node (*node*).

max_depth Limit to the depth of the longest branch on a tree.

min_child_weight Necessary instance (hessian) weight threshold for a split to occur.

eta This parameter refers to the learning rate of the algorithm, a constant used for step size shrinkage.

num_parallel_tree Number of trees to grow in the random forest of a single iteration.

gamma A split will be performed on a node if the predictive power of both its children outweighs the predictive power of the unchanged node by an amount larger than gamma.

rate_drop While **skip_drop** represents the probability that a dropout is performed on an iteration, **rate_drop** represents the fraction of trees that is chosen to be dropped, uniform to all trees in the model.

3.2. Genetic Algorithm

While genetic algorithms have been proposed in the context on GWAS, for the purpose of identifying epistatic interactions, the main approach seems to use the SNPs themselves as part of the building chromosomes. In our implementation, we instead apply the genetic algorithm optimization to the set of parameters discussed in the previous section in order to reach configuration optima for detecting SNP interactions of different sizes. In this work, the tuned algorithm is XGBoost, which acts as a filtering stage to select a subset of features to analyze, followed by an exhaustive search with an objective function — the K2 score.
for each trial run of XGBoost with a specific combination, the GA collects information on how it performed in discovering interactions of interest. The earliest accessible metric is given by XGBoost when it, at each iteration, evaluates the error of the grown boosted tree model. From the exhaustive stage comes the other metric, the minimum K2 score found on all possible combinations among the top ranked features from the resulting XGBoost model by order of combination. A matrix with the collected information is then built as input to a function that will, from that information rank the models according to an user score and thus select which ones will have their information carried over to the next generation.
In a separate structure, the algorithm keeps the chromosomes of the best scored models across all

generations, whether by their lowest K2 Score of a specific order or testing set classification error. This structure will provide the values of the parameters used to build the best configurations, to then use in the creation of the next generation. Some tested priority functions for building the next generation are:

`2_best` This function will crossover the best individuals from all previous generations according to the K2 score by order of interaction with the best individuals from the current generation following the same criteria;

`err_best` This function will crossover instead the all-time best testing set classification error with the best individuals of the current generation by lowest K2 score across all tested combinations per order;

`distn_best` These functions find across the current generation which individuals have the furthest configurations from the all time best scoring individuals, using vector p-norms.

Repeated picks for the crossover operations are to be expected so we then add a mutation to one of the genes per chromosome. For most of the parameters, the mutation is just a random small quantity to add to the original value, taken from an uniformly random distribution and ensuring that it doesn't surpass the previously stated optimal ranges for the algorithm. With the intention of ensuring convergence on the algorithm, the mutations are limited to one per chromosome and crossover happens on a single point, at the middle of the chromosome.

A threshold on the classification error on the testing set is imposed to limit which configuration results will be analyzed in the exhaustive stage. A configuration is rejected further analysis if its classification error on the testing set surpasses 45% and its parameter information will not be used in the construction of the chromosomes in the new generation.

3.3. Feature Importance

To use XGBoost as a filtering stage, we must gather information from the model built to obtain a ranking of all features used to grow the trees in it, according to their importance on the process, and then pick the top ranked, whether by a threshold the importance of a feature must reach or a specified number of features. XGBoost offers several metrics, all based on three attributes the features used to grow the tree possess:

gain The contribution of every split involving that feature;

cover The number of observations whose leaf node was decided based on that feature;

weight The number of times a feature is used to split the data across all trees in the model.

From these metrics, the **gain** is the most relevant to quantify the relative importance of a variable.

3.4. Exhaustive Search

From the objective functions discussed in the state-of-the-art, two possible candidates can be considered for the subspace search, which can involve up to millions of repeated executions of the function, the Bayesian K2 Score and the Mutual Information Score. For an informed choice on which function to use for this stage, we ran exhaustive searches on a toy dataset with 100 SNPs and $1,000$ patients, hiding an interaction with 85% classification accuracy on the first 10 SNPs. For both scores, we did a search on the whole dataset for all combinations of orders 2, 3 and 4.

For figure 6, we show in the x-axis, for each SNP,

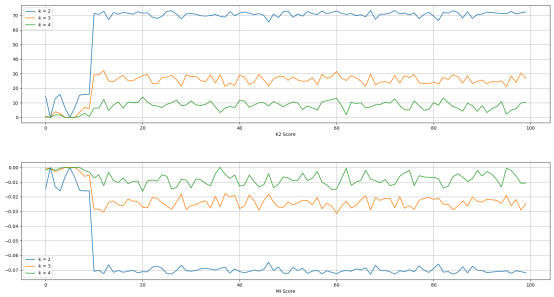

Figure 6: Comparison of exhaustive stage objective functions

the best score it got on all possible combinations of the respective order involving the said SNP. For the MI score, the best score is the highest and for the K2 score the lowest. On the y-axis, we show the difference between the best score across all combinations featuring that SNP and the best overall score for each k. On a absolute scale, the absolute distance of the combinations greatly favours the K2 score. As the order of the tested interactions increases, the less discernible the best combinations will be from the rest, no matter the objective function used. As

Order	# Combinations	MI Score	K2 Score
k = 2	4950	0.277	0.050
k = 3	161700	4.115	2.088
k = 4	3921225	117.574	72.054

Table 1: Time elapsed in seconds for a complete exhaustive search on the same dataset

we can observe from table 1, the deciding factor on choosing the K2 score over MI was the computation time where, on an identical environment, K2 managed to process all interactions faster than MI, no matter the order.

4. Results

To better make an assessment on which type of data each configuration of a parameter better performs, datasets of different size (number of SNPs and number of patients), hidden interactions of different number of SNPs involved (k), and overall accuracy of the interaction in predicting the classification were created. As GWAS datasets usually have significantly more variables due to the size of the human genome than individuals, these datasets were created in a way to mimic those sizes to limit the parameter assessments to the tasks we aim to use the algorithm for.

1. Specify desired size, desired interactions (or just the k of the desired interactions, which will then be created on randomly chosen columns with randomly chosen states), desired accuracy of the interactions on the classification of the model and desired case to control ratio;
2. Initialize a NumPy matrix $\mathbf{X}$ with the desired size and random integer values between 0 and 2 from an uniform distribution;
3. Initialize a NumPy matrix $\mathbf{Y}$, with the same row size as $\mathbf{X}$ and column size equal to the number of interactions to hide in the datasets, with binary values sampled from a distribution that uses the desired case to control ratio (normalized with respect to the column size), which means that the actual case to control ratio will only approximate the desired one;
4. Iterate along the rows of $\mathbf{X}$ and according to the corresponding binary values of $\mathbf{Y}$, change the column values of each iteration to match the state of it on $\mathbf{Y}$. If for that interaction on that row, the value of $\mathbf{Y}$ is true all values of $\mathbf{X}$ on that are changed to the interaction defined states if needed. If false and all values match the interaction states one of them is chosen at random and changed to remove that interaction in the desired control;
5. Reduce $\mathbf{Y}$ to a single-column array by performing an and operation along the rows to get the expected case-control classification;
6. Create another binary column array with the same size as $\mathbf{Y}$ sampled this time from a distribution based on the desired accuracy that will represent which rows in $\mathbf{Y}$ will be changed to false classifications and xor that array with $\mathbf{Y}$ to obtain the actual classifications. The accuracy will then be an approximation of the desired value.

When testing the cooperation of both filtering and exhaustive stages together, we settled on a value of 25 for the parameter top_feats, the number of top ranked SNPs, according to the importance ranking function, to exhaustively analyze. This value, coupled with an exhaustive search from $k = 2$ up to $k = 6$, was chosen to bring the execution time of the exhaustive stage to the same magnitude of that of the filtering stage. The worst-case scenario of the exhaustive search is $\mathcal{O}(\binom{top\_feats}{max\_k}(3^{max\_k} + ind \cdot max\_k))$, where max_k is the higher k analyzed at this stage and ind the number of configurations produced by the GA for the filtering stage at that generation. In figure 7, we can observe how

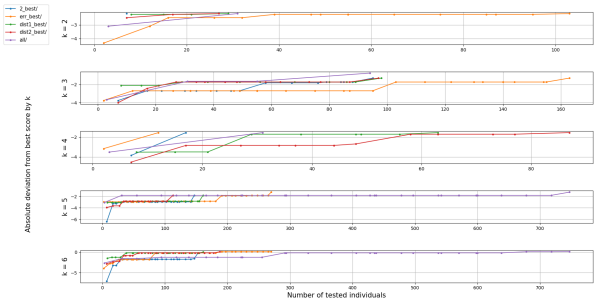

Figure 7: Evolution of the best scoring combinations for different crossover methods in a toy dataset with a hidden $k = 6$ interaction with 70% accuracy

the different crossover methods compared to each other on the same dataset with $1,000$ SNPs and 100 patients. The number of SNPs was chosen to allow an exhaustive search of k up to 5 in sustainable time. The full exhaustive search took over 25 hours. With said exhaustive search, we can compare for $k = \{2, 3, 4, 5\}$ the best scoring interaction discovered by all individuals on each generation with the score of the overall best scoring interaction on the dataset, known beforehand and corresponding to the value 0 in the y-axis, for those orders.
For $k = 6$, an exhaustive search would take years to complete and thus become unreasonable for the purpose of this work. We decide to compare instead the best scoring combinations to another reference, coming from the toy dataset used in the analysis. We test the K2 score of all possible subsets of $k = 6$ from that interaction and choose the best score as reference.
On this dataset, where the interaction hidden was between 6 SNPs and had a prediction accuracy of 70%, we can observe that, while the optimal value was not reached, we could notice improvements on the scores not only at the beginning of the execution but also after several generations, especially on higher orders. The dots serve to separate different generations, being far apart on the *all* GA run due to that method being responsible for an offspring four times bigger than the other methods. The number of offspring also changes from generation to generation, as noted from the variable proximity of the dots for all methods. This phenomena is due to the threshold we placed for the classification er-

ror a model must not reach so it can be exhaustively searched in the follow-up stage. As expected, the *all* method is more successful in obtaining better scores, being a mixture of the other four methods, outscoring the other methods on higher orders on late generations. However, *dist*2 showed a quicker score improvement, especially on $k = 6$.

On a dataset with a hidden interaction of 10%

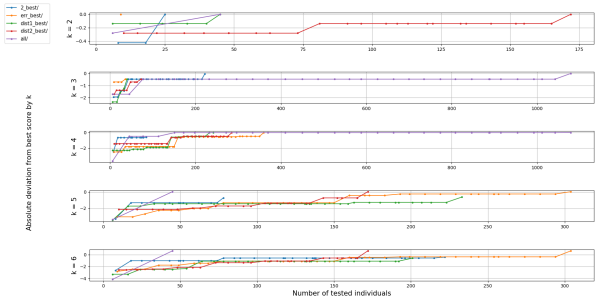

Figure 8: Evolution of the best scoring combinations for different crossover methods in a toy dataset with a hidden $k = 6$ interaction with 80% accuracy

higher accuracy, the algorithm significantly improved its performance, as observed on Figure 8. The best combination was met for k up to 4 by the *all* method and by all methods for $k = 2$. By analyzing the dot concentration and the number of individuals, we can notice that relatively to the 70% accurate dataset, more individuals per generation were admitted to the exhaustive stage. For $k = 3$ the optimal solution was not met early on, as for other orders, but after 30 generations.

Again, after *all*, *dist*2 was the method that had the best performance, reaching solutions of equal score for all k but 3 and 4, despite having a slower improvement over generations than other metrics. The optimal solution was not found for any metric in the 50 generations run for the $k = 5$ analysis. From the quick initial improvement for *all* and the subsequent stagnation, we can draw the conclusion that the algorithm found, for that order and method, a local optima.

For a dataset with a hidden interaction of 90%, our algorithm was able to find all optimal solutions for $k \leq 5$, as seen on Figure 9. Furthermore, the optimal result was met on the first generation for all methods on $k = 2$ and all but *dist*2 on $k = 3$, which found it on the following generation. The *err* method was also able to find the optimal solution for $k = 4$ and fared the best overall, indicating that its performance improves significantly on datasets where the hidden interaction has a good accuracy. *all* again managed to find the best solution for all orders below 6 and equate *err* in solution score at that order. For higher orders, the *dist* methods

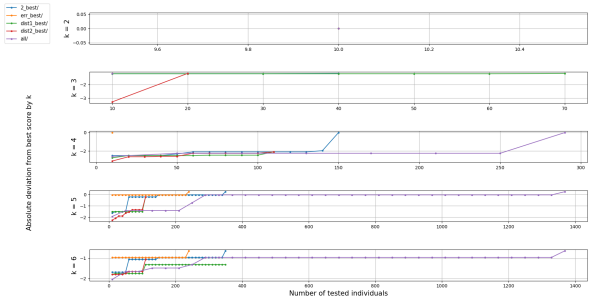

Figure 9: Evolution of the best scoring combinations for different crossover methods in a toy dataset with a hidden $k = 6$ interaction with 90% accuracy

seemed to compete poorly with the other methods. For a better understanding on how each model contributes individually for the best score of a generation, Figure 10 shows the best score found for each tested chromosome (the dots) for the algorithm run on a dataset with a hidden interaction of $k = 6$ with 100% accuracy when determining the disease trait. For methods like 2_*best*, we can observe how, up to the generation the optimal score was found, different models returned different scores. Only after the model scores started to converge to the optimal solution, no pattern or trend being able to discern from them before.

dist methods show the biggest variance between model best scores in a single generation, as shown on $k = 6$. Even after the optimal solution was found and offspring generated using information from the parameter configuration of XGBoost that lead to it, some of these models can only score a K2 score worse by over 10.

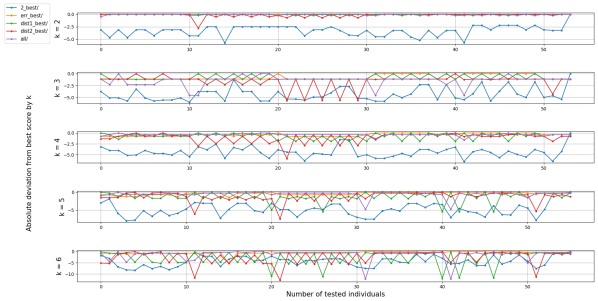

Figure 10: Evolution of the best scoring combinations per XGBoost model for different crossover methods in a toy dataset with a hidden $k = 10$ interaction with 100% accuracy

5. Conclusions

The proposed approach joins the configuration liberty of using a machine learning algorithm as a fil-

tering stage before an exhaustive search with the heuristic power of genetic algorithms, using a multi-objective fitness optimization, to encouraging results. Optimal solutions for k up to 5 where found or closely approached on toy datasets manipulated to include sufficiently accurate interactions. Due to simultaneous improvement of different k solutions, we can conclude that making the algorithm search the best interactions on a range of k, as opposed to a single-objective search, brings benefits to the overall search. Fit genetic individuals for a certain interaction order can lead to an improvement in the solution not only in that order, as the exhaustive stage searches for all k in the desired range.

By providing a viable alternative to a full exhaustive search with a complex framework that uses its retained information to continuously improve itself, producing satisfying results, we designed an approach of considerable utility for detecting epistasis without a restriction for interaction order. Optimal solutions for k where exhaustive search would be time restrictive were found on few generations and the alggorithm was able to escape local optima and show improvements after numerous generations of stagnation in the best solution, using the top performing *all* and *dist*2 crossover methods.

With this work we provided a useful inference tool for GWAS that works outside of the common interaction orders studied, being able to find promising interactions for further biological study without the combinatorial explosion restriction of exhaustive or depth-first searching algorithms.

References

[1] Xgboost parameters.

[2] Single-nucleotide polymorphism — learn science at scitable, Nov 2015.

[3] A. e. a. A. A global reference for human genetic variation. *Nature*, 526:68, 10 2015.

[4] H.-H. Chou, H.-C. Chiu, N. F. Delaney, D. Segrè, and C. J. Marx. Diminishing returns epistasis among beneficial mutations decelerates adaptation. *Science*, 332(6034):1190–1192, 2011.

[5] O. Combarros, M. Cortina-Borja, A. D. Smith, and D. J. Lehmann. Epistasis in sporadic alzheimer's disease. *Neurobiology of aging*, 30(9):1333–1349, 2009.

[6] S.-S. Dong, S. Yao, Y.-X. Chen, Y. Guo, Y.-J. Zhang, H.-M. Niu, R.-H. Hao, H. Shen, Q. Tian, H.-W. Deng, et al. Detecting epistasis within chromatin regulatory circuitry reveals cand2 as a novel susceptibility gene for obesity. *International Journal of Obesity*, page 1, 2018.

[7] P.-J. Jing and H.-B. Shen. Macoed: a multi-objective ant colony optimization algorithm for snp epistasis detection in genome-wide association studies. *Bioinformatics*, 31(5):634–641, 2015.

[8] B. Lehner et al. Combinatorial genetic analysis of a regulatory network reveals the importance of higher order epistasis for gene deletion phenotypes. *bioRxiv*, page 589606, 2019.

[9] P. M Visscher, W. G Hill, and N. Wray. Visscher pm, hill wg, wray nr. heritability in the genomics era-concepts and misconceptions. nat rev genet 9: 255-266. *Nature reviews. Genetics*, 9:255–66, 05 2008.

[10] D. P. McGovern, J. I. Rotter, L. Mei, T. Haritunians, C. Landers, C. Derkowski, D. Dutridge, M. Dubinsky, A. Ippoliti, E. Vasiliauskas, et al. Genetic epistasis of il23/il17 pathway genes in crohn's disease dermot. *Inflammatory bowel diseases*, 15(6):883–889, 2009.

[11] N. Modjtahedi, E. Hangen, P. Gonin, and G. Kroemer. Metabolic epistasis among apoptosis-inducing factor and the mitochondrial import factor chchd4. *Cell Cycle*, 14(17):2743–2747, 2015.

[12] J. H. Moore and S. M. Williams. Traversing the conceptual divide between biological and statistical epistasis: systems biology and a more modern synthesis. *BioEssays*, 27(6):637–646, 2005.

[13] R. Nunkesser, T. Bernholt, H. Schwender, K. Ickstadt, and I. Wegener. Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics*, 23(24):3280–3288, 2007.

[14] K. V. Rashmi and R. Gilad-Bachrach. Dart: Dropouts meet multiple additive regression trees. *ArXiv*, abs/1505.01866, 2015.

[15] M. D. Ritchie, L. W. Hahn, N. Roodi, L. R. Bailey, W. D. Dupont, F. F. Parl, and J. H. Moore. 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, 2001.

[16] P. C. Ruiz. Illustration of the k2 algorithm for learning bayes net structures.

[17] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[18] S. Tuo, J. Zhang, X. Yuan, Z. He, Y. Liu, and Z. Liu. Niche harmony search algorithm for detecting complex disease associated high-order snp combinations. In *Scientific Reports*, 2017.

[19] S. Tuo, J. Zhang, X. Yuan, Y. Zhang, and Z. Liu. Fhsased: Two-locus model detection for genome-wide association study with harmony search algorithm. *PLOS ONE*, 11(3):1–27, 03 2016.

[20] J. C. Turton, J. Bullock, C. Medway, H. Shi, K. Brown, O. Belbin, N. Kalsheker, M. M. Carrasquillo, D. W. Dickson, N. R. Graff-Radford, et al. Investigating statistical epistasis in complex disorders. *Journal of Alzheimer's Disease*, 25(4):635–644, 2011.

[21] T. Van de Cruys. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, DiSCo '11, pages 16–20, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[22] D. M. Weinreich, Y. Lan, C. S. Wylie, and R. B. Heckendorn. Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development*, 23(6):700–707, 2013. Genetics of system biology.

[23] J. Wu, B. Devlin, S. Ringquist, M. Trucco, and K. Roeder. Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society*, 34(3):275–285, 2010.

[24] C. Yang, Z. He, X. Wan, Q. Yang, H. Xue, and W. Yu. Snpharvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies. *Bioinformatics*, 25(4):504–511, 2008.