Evaluation of Existing Methods for High-Order Epistasis Detection

Christian Ponte-Fernández, Jorge González-Domínguez, Antonio Carvajal-Rodríguez, and María J. Martín

Abstract—Finding epistatic interactions among loci when expressing a phenotype is a widely employed strategy to understand the genetic architecture of complex traits in GWAS. The abundance of methods dedicated to the same purpose, however, makes it increasingly difficult for scientists to decide which method is more suitable for their studies. This work compares the different epistasis detection methods published during the last decade in terms of runtime, detection power and type I error rate, with a special emphasis on high-order interactions. Results show that in terms of detection power, the only methods that perform well across all experiments are the exhaustive methods, although their computational cost may be prohibitive in large-scale studies. Regarding non-exhaustive methods, not one could consistently find epistasis interactions when marginal effects are absent. If marginal effects are present, there are methods that perform well for high-order interactions, such as BADTrees, FDHE-IW, SingleMI or SNPHarvester. As for false-positive control, only SNPHarvester, FDHE-IW and DCHE show good results. The study concludes that there is no single epistasis detection method to recommend in all scenarios. Authors should prioritize exhaustive methods when sufficient computational resources are available considering the data set size, and resort to non-exhaustive methods when the analysis time is prohibitive.

Index Terms—Detection power, high-order epistasis, false positives, genetic interaction, review, survey

1 Introduction

An important challenge in genetic medicine, and in genetics in general, is the correct assessment of the genetic basis of a disease or phenotypic effect. Current Genome-Wide Association Studies (GWAS) analyze data sets comprised of hundreds of thousands of genetic markers genotyped for thousands of individuals. However, despite this huge amount of information, our understanding of the genetic architecture of complex traits and diseases is still limited [1]. The identification of the genetic cause of some traits and diseases may be hindered, among others, by epistasis. Originally, epistasis was defined as the interaction of two or more loci for a specific phenotype [2] so that the effect of a mutation can be different depending on the genetic context. In the genomic era, epistasis may involve the interaction of different loci and/or different markers within the same loci. If epistasis involves more than two loci it is called high-order epistasis [3]. Epistasis has important evolutionary implications with an impact in gene prediction, molecular evolution and infectious diseases. It may also have an effect on the evolution of drug resistance as antibiotic resistance [3] and HIV drug resistance [4]. Understanding how mutations in pathogens interact should improve the prediction of pathogen evolution and vaccine development. Epistasis is also important in personalized medicine and biotechnology, and can improve protein design by informing about protein structure and interaction.

Most genetic studies are not able to detect high-order epistasis despite possibly being present in many proteins, from viral to mammalian, thus making it difficult to determine its importance in heritable phenotypes. Detecting the high-order interactions in a genome-wide scale implies the computational challenge of evaluating the huge number of loci combinations plus the statistical challenge of a high dimensional problem [5]. Therefore, the fact that most reported genetic interactions involve only two loci is due to technical limitations rather than the underlying biology [6], [7], [8].

Prior to this work, there have been several review studies that compared different strategies for epistasis detection from various perspectives. Some are focused entirely on their methodology, comparing the different approaches, their advantages and limitations [9], [10], [11], [12], [13], [14], [15]. Other studies go further by also including an empirical comparison from simulation studies, although the number of methods included in these studies is more limited [16], [17], [18], [19], [20]. There are also previous publications regarding the selection of epistatic detection methods and how to integrate them in the different stages of a genetic study [21], [22], [23]. Nevertheless, there is no previous comparison study with an emphasis on the interaction order.

In this work, we compare the epistasis detection methods published during the last ten years with a special interest in high-order interaction detection. To accomplish that, we have selected those methods that, first, support epistasis detection for qualitative phenotypes and for more than two loci (in the form of Single Nucleotide Polymorphisms, or, in

- *Christian Ponte-Fernández, Jorge González-Domínguez, and María J. Martín are with the Universidade da Coruña, CITIC, Computer Architecture Group, 15001 A Coruña, Spain. E-mail: {christian.ponte, jgonzalezd, mariam}@udc.es.*
- *A. Carvajal-Rodríguez is with the Departamento de Bioquímica, Genética e Inmunología y Centro de Investigación Mariña (CIM), Universidade de Vigo, 36310 Vigo, Spain. E-mail: acraaj@uvigo.es.*

Manuscript received 14 Apr. 2020; revised 12 Aug. 2020; accepted 5 Oct. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Christian Ponte-Fernández.)
Digital Object Identifier no. 10.1109/TCBB.2020.3030312

TABLE 1
Alphabetically Sorted List of the Different Methods Included in this Work, Together With the Strategy Followed, the Implementation Language Used, the Year That They Were Published and Their Websites

Method	Strategy	Language	Year	URL	Ref.
AntMiner	Swarm intelligence	MATLAB	2012	https://sourceforge.net/projects/antminer/	[24]
ATHENA	Genetic Algorithm	C++	2010	https://ritchielab.org/software/athena-downloads	[25]
BADTrees	Depth-first	C++	2012	https://github.com/guyrt/WFUBMC	[26]
BEAM3	Random-search based	C++	2012	http://personal.psu.edu/yzz2/software/	[27]
BHIT	Random-search based	C++	2015	http://digbio.missouri.edu/BHIT/	[28]
CINOEDV	Swarm intelligence	R	2016	https://github.com/cran/CINOEDV	[29]
DCHE	Filtering	Java	2014	http://www.cse.unt.edu/~xuanguo/project_dche.html	[30]
EACO	Swarm intelligence	MATLAB	2018	https://sourceforge.net/projects/eaco1/	[31]
EDCF	Filtering	C/C++	2012	http://www.cs.ucr.edu/~minzhux/EDCF.zip	[32]
epiACO	Swarm intelligence	MATLAB	2017	https://sourceforge.net/projects/epiaco1/	[33]
EpiMiner	Filtering	MATLAB	2014	https://sourceforge.net/projects/epiminer/	[34]
FDHE-IW	Depth-first	MATLAB	2018	https://www.mdpi.com/2073-4425/9/9/435\#supplementary	[35]
GALE	Genetic Algorithm	Python	2010	http://gbml.org/2010/06/10/python-lcs-implementations-gale-gassist-for-snp-environment/	[36]
HiSeeker	Filtering	C++	2017	http://mlda.swu.edu.cn/data.php	[37]
IACO	Swarm intelligence	MATLAB	2016	https://sourceforge.net/projects/iaco1/	[38]
LAMPLINK	Filtering	C++	2016	https://github.com/a-terada/lamplink/	[39]
LRMW	Depth-first	C++	2014	https://msu.edu/~qlu/Software.html	[40]
MACOED	Swarm intelligence	C++/MATLAB	2015	http://www.csbio.sjtu.edu.cn/bioinf/MACOED/	[41]
MDR	Exhaustive	Java	2001	https://multifactordimensionalityreduction.org/	[42]
MECPM	Filtering	C	2009	https://www.cbil.ece.vt.edu/ResearchOngoingSNP.htm	[43]
Mendel	Filtering	C/C++	2009	http://software.genetics.ucla.edu/download?package=1	[44]
MPI3SNP	Exhaustive	C++	2019	https://github.com/chponte/mpi3snp	[45]
NHSA-DHSC	Swarm intelligence	MATLAB	2017	https://www.nature.com/articles/s41598-017-11064-9\#Sec28	[46]
SingleMI	Filtering	C++ & CUDA	2017	https://github.com/sleeepyjack/singlemi/	[47]
SNPHarvester	Random-search based	Java	2009	http://bioinformatics.ust.hk/SNPHarvester.html	[48]
SNPRuler	Depth-first	Java	2010	http://bioinformatics.ust.hk/	[49]
StepPlr	Depth-first	R	2008	https://cran.r-project.org/web/packages/stepPlr/index.html	[50]

short, SNPs); second, offer an implementation freely available to the scientific community and finally, their execution can be completed within a week. Table 1 lists all methods included. We decided to also consider MDR and StepPLR, despite being published more than ten years ago, due to their relevance in the field. For each method, its detection power and error rates were measured using more than 5,000 synthetic data sets, each one involving different simulation conditions in order to make a fair comparison.

2 Methods

This section provides a brief description of the selected methods to highlight their similarities and differences, and to have a better understanding of the results that each program yields. We refer to the authors' original works for a more complete and in-depth explanation. The selected methods have been grouped into six categories, attending at how the search space is explored: exhaustive methods, filtering methods, depth-first methods, swarm intelligent methods, genetic algorithms and random-search-based methods.

2.1 Exhaustive Methods

Exhaustive methods apply the brute force technique to the association search problem, exploring all possible combinations of genetic markers up to a defined size or order. The computational cost of exploring all possible combinations is exponential with the number of genetic markers considered and the combination size. Therefore these methods cannot be applied to large data sets with high epistatic factors.

MDR [42] and MPI3SNP [45] fall under this category. MDR partitions the individuals in the data set into different k-fold cross-validation groups. Combinations are evaluated through a prediction model which labels the different allele combinations as high-risk (if the number of cases exceeds the number of controls for that particular combination) or as low-risk (if it does not). For each combination, k different models are created (one per cross-validation partition) and its prediction accuracy is averaged across partitions. At the end of MDR, the combination corresponding to its best-averaged prediction accuracy is reported. MPI3SNP, instead, enumerates all third-order combinations and sorts them using Mutual Information, returning the top-ranked ones. The version of MPI3SNP used in this study is a modification of the tool described in [45], allowing the user to specify the order of the combinations explored.

2.2 Filtering Methods

Filtering methods discard a large number of SNPs or combinations of SNPs to reduce the computational burden. The most direct approach is to filter the individual SNPs of the data set before attempting to combine them, drastically decreasing the number of combinations. EpiMiner [34] and Mendel [44] follow this approach. EpiMiner ranks individual SNPs by their Co-Information Index (CII) and retains the top ranked ones. The number of retained SNPs can be fixed or selected on a case-by-case basis through a Support Vector Machine (SVM). The retained SNPs advance to a second stage where all possible combinations among them are evaluated using permutation-based Co-Information, and combinations whose p-values surpass a certain threshold are reported as interactions. Computing the Co-Information Index requires calculating the index for all the combinations which contain a certain SNP up to a certain order, which

still supposes a costly step, therefore EpiMiner allows us to approximate its value through Monte Carlo sampling. Mendel uses a lasso penalized logistic regression model to quantify the association between the SNPs, used as predictor variables, and the phenotype, used as the regression class. The interaction search process begins by pre-screening the SNPs in the data set in a first stage using a simplified regression model and an absolute score criterion. Then, the number of SNPs selected is further reduced by tuning the constant λ, which increases the lasso penalty and, in turn, leaves many predictors out of the logistic regression model. Finally, when the number of retained SNPs is very small, the penalty is removed and the model coefficients are re-estimated. Using this final model, p-values of individual and combinations of SNPs are assessed following a leave-one-out procedure and thus the associated combinations are identified.

Other methods perform the filtering step on low-order combinations. HiSeeker [37] and MECPM [43] enumerate all possible 2-SNP combinations and select a group of candidates for further analysis. HiSeeker filters these combinations by applying Pearson's χ^2 test with eight degrees of freedom, assessing the association between each combination and the phenotype. Combinations that meet a relaxed Bonferroni-corrected p-value threshold proceed to a second stage for a higher-order analysis. HiSeeker offers the possibility of performing an exhaustive search during the second stage to find high-order interactions, or using an Ant Colony Optimization (ACO) algorithm if the number of combinations to be tested is still unreasonably high. ACO algorithms will be covered in detail in Section 2.4. In the end, the non-relaxed Bonferroni-corrected p-value threshold is used to filter false positives. MECPM creates a maximum entropy classification model and uses the Bayesian Information Criterion (BIC) to quantify the association between genotypes and the phenotype under study. For this purpose, MECPM first creates a pool of promising SNP combinations and iteratively adds combinations to the model until the BIC cost is minimum. The pool is constructed following two approaches: a complete approach where all single SNPs and combinations of two SNPs serve as seeds, and successive SNPs are appended to each seed measuring the change in BIC cost with each addition; and a greedy approach where the initial selected seeds are reduced to the top-ranking single and combinations of two SNPs using the relative entropy, and successive SNPs are appended maximizing this metric. MECPM reports the SNP combinations included in the model.

DCHE [30], EDCF [32] and SingleMI [47] use clustering techniques to filter combinations of SNPs. Both DCHE and EDCF recursively apply a clustering algorithm over the population frequencies of all allele combinations, starting from 2-SNP combinations up to a selected order. These clusters are then tested using Pearson's χ^2 test to measure its association with the phenotype. DCHE implements a clustering algorithm named *Dynamic Clustering* which reduces the 3^k frequencies associated with a combination of k SNPs in a biallelic population to a number between 3 and 6, merging the two least significant allele combinations in each step. DCHE retains a different fixed number of top-ranking combinations depending on the combination order being explored and applies a p-value threshold at the end of the algorithm to filter out irrelevant combinations. EDCF, instead, creates three groups from all allele combination frequencies: G_0, or combinations which occur more frequently in cases than in controls; G_1, or combinations which occur more frequently in controls than in cases; and G_2 with the combinations left. Clusters are then evaluated using a permutation test and the corresponding SNP combination is discarded if their p-value does not meet a certain threshold. Again, a fixed number of top-ranking SNP combinations (using the aforementioned χ^2 test) are retained from each combination size and its Bonferroni-corrected p-value is finally used as the threshold to decide the result of the method. SingleMI uses a clustering algorithm in a very different manner from the previous two. Individual SNPs are clustered following a K-Means clustering method, where the distance between SNPs and the centroid of each cluster is measured using Mutual Information. Markers that are strongly interacting pair-wise tend to be placed in different clusters. Therefore, after creating the K clusters, a user-defined number of SNPs from different clusters are analyzed exhaustively using the same Mutual Information metric.

LAMPLINK [39] follows a completely different filtering approach from previous methods. Individual SNP genotypes are first categorized into two classes following a dominant or recessive exclusive model: risk and non-risk classes. Then, a modified version of the pattern mining algorithm called Linear time Closed itemset Miner (LCM) [51] is used to prune the SNPs combinations that, taking into account their frequency, cannot show a significant association with the phenotype. Finally, the non-pruned combinations are evaluated using a Fisher's exact test or a chi-squared test and the obtained p-value is corrected according to the number of testable combinations.

2.3 Depth-First Methods

This group is made of methods that explore the combination space using a depth-first search method, incorporating SNPs on each iteration while maximizing some measurement until convergence is detected. This search is repeated successively until a certain number of combinations is reached or no more significant combinations can be found. FDHE-IW [35], LRMW [40], BADTrees [26], StepPLR [50] and SNPRuler [49] follow this procedure.

FDHE-IW implements a search algorithm which constructs SNP combinations incrementally, starting with the empty set and repeatedly adding the SNP that maximizes the Symmetrical Uncertainty of the set until a maximum set size is reached. A G-Test is applied after achieving a number of combinations to obtain a p-value associated with the combinations. LRMW uses decision trees to represent candidate interactions and employs its associated Area Under the ROC Curve (AUC) to measure significance. The method starts with an empty tree and progressively generates more complex ones until an AUC value of 1 is reached. Then, a 10-fold cross-validation is carried out to select the most complex model which still improves the AUC compared to the previous one. Decision trees are also used in BADTrees to represent interaction among SNPs and a method called bagging is introduced to increase the signal-to-noise ratio of the interacting SNPs. Bagging consists in bootstrapping a

number of data sets from the original one, constructing a tree in each of the sets and finding similarities among them. In BADTrees, the most frequent SNPs among the trees are reported as associated with the phenotype.

StepPLR uses a penalized logistic regression model to quantify the association between the selected SNPs and the phenotype. It is an iterative algorithm where, based on a cost-complexity statistic which integrates either the Akaike Information Criterion or the BIC, SNPs or combination of SNPs are added or removed from the model in a series of forward selection and backward deletion steps. The model with the minimum cost is selected and the SNPs or combinations of SNPs included in the model are reported. Lastly, SNPRuler uses a rule-based classification model which introduces the concept of rule utility and its derived upper bound to identify whether a rule can be further improved to increase its classification accuracy or not. SNPRuler begins by building a search tree to guide the search of interactions, where nodes represent SNPs and edges represent interactions between SNPs. The tree is built avoiding unnecessary expansions of child nodes, i.e., those whose utility's upper bound is lower than a certain threshold or its parent's utility. After the search tree is built, SNPRuler finds a number of top-ranked interactions (paths from the root to the leaf nodes) sorted by its utility measurement, calculates their p-value using the χ^2 statistic and writes the list to an output file.

2.4 Swarm Intelligent Methods

Swarm intelligence (SI) is a group of methods that falls under the category of metaheuristics. Metaheuristics are high level heuristic methods for exploring the search space, applicable to domains where the computational power of the information systems is insufficient, or the domain information is limited [52]. Swarm intelligence, as many of the metaheuristics, are nature-inspired methods that rely on the problem-solving ability that emerges from the interactions of simple information-processing units, or agents [53]. These are multi-agent, decentralized and self-organized systems where the individual agents that integrate the system follow a rule-set that determines their behaviour.

ACO is the most explored metaheuristic in epistasis detection. It relies on artificial ants (independent decision-making agents) to iteratively explore the SNP combination space. Pheromones are an implicit communication mechanism that ants use to guide the search. Whenever an ant explores a combination, it deposits a certain number of pheromones proportional to the association strength between the phenotype and the specific combination. Pheromones also evaporate over time, progressively reducing its effect. A probability function is used to decide which combination an ant should explore next based on the pheromone levels present on the combinations. The probability function also considers selecting a random combination under specified odds to avoid being trapped in local optima. After a fixed number of iterations are completed, the algorithm ends, and the result is a list of the most promising combinations visited by the ants. MACOED [41], IACO [38], epiACO [33] and HiSeeker [37] implement this method faithfully, only exchanging the association measure and how the results are treated. MACOED uses the Pareto Optimal Set to select a group of candidate combinations from all explored and then applies a Pearson's χ^2 test to quantify its association. IACO and epiACO use the ratio between the Mutual Information and the Bayesian Network, and the ratio between the Mutual Information and the K2 score, respectively, to measure association. Both methods then proceed to calculate an inflexion point on the association value to separate significant from irrelevant combinations. HiSeeker, as explained in Section 2.2, runs the ACO algorithm on a filtered group of SNPs. It uses Pearson's χ^2 test to evaluate the association, and the top-ranked combinations reported by the ACO algorithm are evaluated using the χ^2 test again to provide a Bonferroni-corrected p-value metric.

AntMiner [24] and EACO [31] innovate over the generic ACO algorithm by incorporating a heuristic into the probability function. AntMiner includes the addition of the Symmetrical Uncertainty and Spatially Uniform ReliefF onto the probability function, and segregates the ants into sub-colonies each exploring combinations of different sizes. It uses Pearson's χ^2 test as the association measurement. All explored combinations that surpass a certain χ^2 threshold are kept in what they call a Candidate Set, which is post-processed at the end to reduce false positives. EACO, on the other hand, uses the Multiple Threshold Spatially Uniform ReliefF as the heuristic of choice, and uses the ratio between Mutual Information and Gini index to assess association. Similarly to IACO and epiACO, significant combinations are identified by calculating an inflexion point on the association metric.

CINOEDV [29] and NHSA-DHSC [46] use different swarm intelligence methods from the extensively seen ACO. CINOEDV implements the Particle Swarm Optimization (PSO) algorithm, where agents consist of particles with a defined position and velocity. The position represents the selected SNP combination, and from each position, its fitness or degree of association with the phenotype can be obtained using three different metrics: Co-Information, Normalized Co-Information and Contribution Co-Information. The velocity of each particle determines the next position to be explored. It depends on the current velocity, the best position found by the current particle and the best global position found by all particles. The algorithm initializes all particles' positions and velocities randomly and iterates for a fixed number of steps, storing the best position found on each iteration. It returns the list of positions sorted by the selected metric. The NHSA-DHSC method consists of two stages, a searching step that implements the Niche Harmony Search Algorithm combining a Harmony Search (HS) algorithm with a niching technique, and a second stage where all found candidates are evaluated. HS is a music-inspired swarm intelligent algorithm that mimics the improvisation process used by skilled musicians, where harmonies representing SNP combinations are iteratively explored following an improvisation process and the best harmonies are kept in a harmony memory [54]. The improvisation of new harmonies consists in choosing between pitch-adjusting previous harmonies and randomly exploring new ones. When the algorithm is stuck in a local optimum the niching algorithm is triggered, and the centroid and radius of the optimum point are included in a taboo table to be avoided by all future solutions, forcing the HS algorithm to explore new areas in the solution space. NHSA-DHSC uses three different association metrics, kept in separate

harmony memories, which are the K2-score, the Gini index and the joint entropy. After the NHSA algorithm ends, the three memories are joined into a common candidate pool and a G-test is performed on the resulting combinations to check for association with the phenotype.

2.5 Genetic Algorithms

Genetic Algorithms (GA) are another group of metaheuristic methods which mimic the biological evolution process. GAs begin with a population of random solutions to a problem, encoded as chromosome-like data structures. The algorithm explores the solution space by evolving the current population into successive generations following a reproductive function. Reproduction consists of evaluation, selection, recombination and mutation steps. Solutions are evaluated using a fitness function, and reproductive opportunities are given proportionally to each individual according to its fitness. Selected individuals create offspring in a recombination operation, in which the two encoded solutions create two new offspring by selecting a (random) recombination point and swapping the subsequent fragments. Finally, a mutation step modifies some bits of the offspring following a specific probability function. The method evolves the population until a certain fitness of the solutions is achieved or the number of generations reaches the limit [55]. GALE [36] and ATHENA [25] use GAs to detect epistatic interactions.

GALE creates a rule-based classification system using a GA to generate a rule set. The solutions of the population are ordered rule sets from which a rule-based classifier can be built. The fitness of a solution is measured as the average accuracy of its classifier in a k-fold cross-validation partition. GALE introduces the concept of spatial awareness to GAs by representing the population of solutions in a 2D grid and modifying the reproductive selection to take into account the proximity between solutions in the grid [56]. The final rule set obtained at the end of the GA is the solution provided by GALE.

ATHENA introduces Grammatical Evolution Neural Networks to the epistasis detection problem. Grammatical Evolution is a GA dedicated to the construction of computer programs, adapting the representation of solutions and the reproductive methods for this purpose. Solutions are variable length binary strings made of groups of 8 bits named codons, each encoding an integer. Codons are translated into rules following a predefined grammar specified in Backus-Naur Form (BNF), and the translation of a complete solution is a program which can be evaluated using a fitness function [57]. ATHENA uses the coefficient of determination, R^2, as the fitness function to evaluate the different solutions considered. These solutions are made up of the SNPs used as input variables to the neural network, the network architecture itself and the weights associated to each of the connections. Using the BNF grammar, the different components of the solutions can be translated into a fully functional neural network. ATHENA also replaces the single-point crossover method from GAs with the Tree-Based Crossover method, which swaps a complete branch of the neural network to create offspring in order to avoid the uncertainty of recombining the network in its binary representation. ATHENA applies a 5-fold cross-validation to construct five different classification models and selects the model whose SNPs appear more consistently as the best model.

2.6 Random-Search-Based Methods

Lastly, a group of methods based on the random search algorithm can be identified. Random search stochastically samples the solution space for a number of iterations, evaluates each solution using a fitness function and saves the result with the best fitness value out of all the explored. SNPHarvester [48], BEAM3 [27] and BHIT [28] are epistasis detection methods that belong to this group.

SNPHarvester implements an algorithm named *PathSeeker* to explore multiple combinations by the means of different local search iterations at random points of the combination space. *PathSeeker* follows a swapping technique, testing for all SNPs if any replacement in the combination can improve the χ^2 association value until no more replacements can be made. Once a predefined number of candidates has been found, a post-processing step is carried out to filter out spurious interactions by fitting a L_2 penalized logistic regression and reporting those interactions selected by the regression model. BEAM3 uses a joint probability model between the SNP collection X, the interacting SNPs X_1 and a disease graph G; and the phenotype Y to determine the association present in the data. G is an undirected graph where nodes represent non-overlapping groups of SNPs from X_1 and edges represent interactions between groups. BEAM3 explores the search space using Monte Carlo Markov Chain (MCMC) sampling to update the selected SNPs in X_1 and its graph representation in G repeatedly. The sampling process adds or removes SNPs in or out of X_1 and updates the nodes and edges of G accordingly. After a number of iterations are completed, the algorithm ends and the best model is returned. BHIT also resorts to a probability model to assess the association between genotypes and a phenotype, but this tool divides the genotype markers into different partitions. BHIT initializes the partition variable I by placing each SNP into a different partition and iteratively samples I using MCMC, maintaining the changes to I between iterations if the probability of the model increases. When the iterative process finishes, BHIT returns the different partitions in which the SNPs have been divided, the interacting SNPs being the ones grouped in the same partition as the phenotype variable.

3 Evaluation

The evaluation section of the different epistasis detection methods is separated into four parts: data simulation design, runtime evaluation, detection power analysis and false positive testing. In data simulation design, the pipeline created for simulating the data sets used in successive subsections is explained in detail. Runtime evaluation briefly compares how the different methods perform in terms of execution time. Detection power measures the ability to locate combinations of SNPs associated with the phenotype under different simulation conditions. Lastly, false positive testing measures the ability to discern between significant and non-significant combinations.

Parameterization of the methods is consistent across the whole evaluation. In general terms, parameter selection was done either using the same values of the evaluation presented in its original work or following indications from the authors. The exception to this rule were swarm intelligent methods, where the number of iterations and agents is common to all methods in order to ensure a fair comparison. For most methods, there is a clear distinction in parameterization for third and fourth-order searches. When there is no interaction in the data, the parameters corresponding to the highest order admissible are selected. Section 1 of the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2020.3030312, covers, in detail, how the different parameters were chosen for each program.

3.1 Data Simulation Design

A large number of data sets were developed for the evaluation of the methods, with varying features from one another in order to model different characteristics of the simulated population. The design goal of the simulation process was to generate a wide variety of data sets resembling real populations, therefore the parameterization used for modelling the population was chosen using estimates from real traits.

The simulation was carried out using GAMETES [58]. In GAMETES, epistatic interactions are described as penetrance tables, which define the penetrance of all possible allele sequences in a specific loci combination. In this study, we considered model-driven interactions showing marginal effects, and model-free interactions with no marginal effects.

Penetrance tables with no marginal effects can be obtained natively through GAMETES, which follows a stochastic generation procedure to find epistatic relationships [58] under no model assumption. Model-driven penetrance tables, on the other hand, cannot be calculated within GAMETES and thus were obtained from Toxo [59], a MATLAB library which can compute penetrance tables from epistasis models. In this study, we employed the widely used additive and threshold models proposed by Marchini *et al*. in [60], two models that define epistatic interactions with marginal effects.

Both GAMETES and Toxo calculate penetrance tables meeting a certain parameterization. The following list describes what these parameters are, and what criteria we used to select values:

- *Minor Allele Frequency*. The frequency at which the second most common allele occurs in a given population. The distribution of observed susceptibility SNPs is skewed towards higher minor-allele frequencies (MAF $>$ 20 percent) [61] and there is an increasing difficulty of detecting disease-causing variants with low MAF [62]. An accepted standard of MAF is 0.1, thus we have assayed values in the range [0.1, 0.4].
- *Heritability*. The degree to which individual genetic variation accounts for the population phenotypic variation [63]. Heritability estimates of human traits for several medical conditions usually cluster in functional domains with its highest values between 70 and 80 percent and the lowest ones between 20 and 30 percent [64]. Therefore, we selected heritability values from the range [0.1, 0.8].
- *Prevalence*. The proportion of individuals from a population that carries a specific trait or suffers from a disease. Diseases can be categorized as rare if their prevalence is under 0.0005 (fewer than 1 in 2,000 people), and ultra-rare if it is under 2E-05 (fewer than 1 in 50 000 people) [65]. For this simulation study, we have restricted prevalence values to be greater than 1E-06.

Table 2 lists all the parameters of the penetrance tables used throughout the evaluation. The criteria were to create penetrance tables of third and fourth-order, with MAF values of 0.10, 0.25 and 0.40 and heritabilities of 0.10, 0.25, 0.50 and 0.80 whose prevalence is above 1E-06. Model-driven tables cannot be obtained for every combination of MAFs, prevalence and heritability due to restrictions in the underlying mathematical model [59], resulting in a different number of tables according to the model. GAMETES, on the other hand, follows a probabilistic approach, which has problems to find model-free tables when increasing the interaction order, decreasing the MAF and increasing the heritability. Consequently, many combinations could not be obtained in a reasonable time.

From each penetrance table, 100 data sets were generated containing 500 SNPs from 2,000 individuals (1,000 cases and 1,000 controls). Non-interacting loci were simulated using a MAF randomly sampled from the interval [0.05, 0.5]. In total, the data collection is comprised of 55 different epistatic relationships, 5,500 data sets, 2.75 million SNPs and 11 million individuals.

Lastly, for the false positive testing, we also simulated 100 data sets with 500 SNPs from 2,000 individuals (1,000 cases and 1,000 controls) containing no interaction. Loci for these data sets were also sampled from the MAF interval [0.05, 0.5].

All the simulation configurations, epistasis models, penetrance tables and data sets are publicly available at Github.[1]

3.2 Runtime Evaluation

The runtime for each of the method's implementation was measured and compared using a single core of an Intel Sandy Bridge 2660 from the Pluton cluster (Supplementary Table S1, available online). SingleMI is the only exception, since it uses NVIDIA GPUs, and thus it was run on an NVIDIA Tesla K20m, also available at the Pluton cluster. Fig. 1 compares the average runtime of all the studied tools for third and fourth-order analyses, across five repetitions. The first data set of the additive model using MAF = 0.25 and heritability = 0.25, both for third and fourth-order, was arbitrarily chosen for this purpose.

MDR, EpiMiner and CINOEDV's runtimes could not be measured due to a restriction on the maximum allocatable time equal to three days. HiSeeker's runtime for fourth-order searches could not be measured as well, due to errors in the program which are not present during third-order searches.

Results show a clear distinction in runtime between exhaustive and non-exhaustive methods: exhaustive methods are largely influenced by the interaction order, while

1. https://github.com/UDC-GAC/epistasis-simulation-data

TABLE 2
Interaction Orders, Minor Allele Frequencies (MAF), Prevalence Values ($P(D)$) and Heritability Values (h^2) of the Penetrance Tables Used During the Data Simulation

Order	MAF	P(D)	h^2
3	0.10	0.000012	0.10
3	0.10	0.000004	0.25
3	0.10	0.000002	0.50
3	0.10	0.000001	0.80
3	0.25	0.005370	0.10
3	0.25	0.001153	0.25
3	0.25	0.000504	0.50
3	0.25	0.000306	0.80
3	0.40	0.254558	0.10
3	0.40	0.022186	0.25
3	0.40	0.008545	0.50
3	0.40	0.005091	0.80
4	0.25	0.000234	0.10
4	0.25	0.000068	0.25
4	0.25	0.000031	0.50
4	0.25	0.000019	0.80
4	0.40	0.036282	0.10
4	0.40	0.003383	0.25
4	0.40	0.001374	0.50
4	0.40	0.000822	0.80

(a) Additive relationship

Order	MAF	P(D)	h^2
3	0.10	0.064602	0.10
3	0.10	0.025561	0.25
3	0.10	0.013270	0.50
3	0.10	0.008417	0.80
3	0.25	0.477516	0.10
3	0.25	0.267707	0.25
3	0.25	0.154539	0.50
3	0.25	0.102529	0.80
3	0.40	0.780354	0.10
3	0.40	0.586967	0.25
3	0.40	0.415395	0.50
3	0.40	0.307526	0.80
4	0.10	0.012563	0.10
4	0.10	0.005140	0.25
4	0.10	0.002590	0.50
4	0.10	0.001623	0.80
4	0.25	0.275518	0.10
4	0.25	0.132034	0.25
4	0.25	0.070683	0.50
4	0.25	0.041819	0.80
4	0.40	0.668428	0.10
4	0.40	0.446405	0.25
4	0.40	0.287337	0.50
4	0.40	0.201273	0.80

(b) Threshold relationship

Order	MAF	P(D)	h^2
3	0.25	0.5860	0.10
3	0.25	0.4923	0.25
3	0.25	0.4223	0.50
3	0.40	0.5163	0.10
3	0.40	0.5644	0.25
3	0.40	0.5019	0.50
3	0.40	0.4970	0.80
4	0.25	0.4201	0.10
4	0.25	0.5910	0.25
4	0.40	0.4720	0.25
4	0.40	0.4356	0.10

(c) Relationship with NME

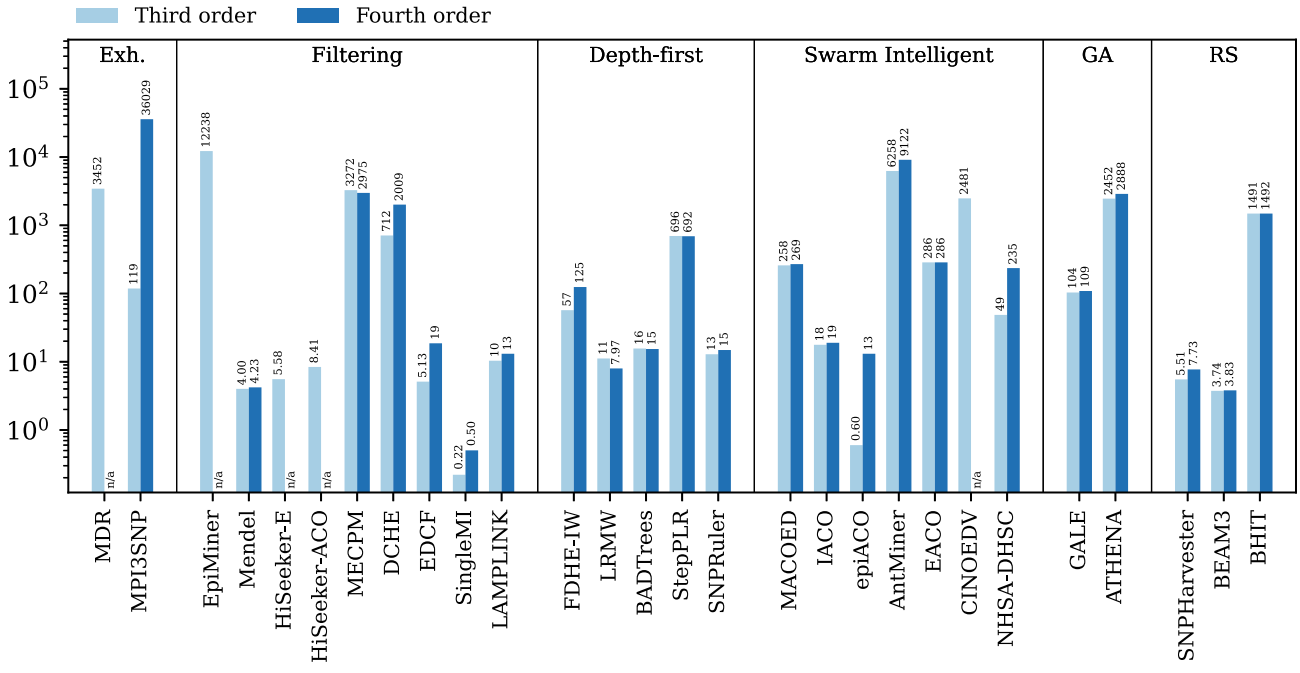


Fig. 1. Average runtime, in seconds, of the different methods for third and fourth order interactions.

non-exhaustive methods generally remain unaffected when moving from third to fourth-order. The only exceptions are EpiMiner and CINOEDV methods, which already show an extraordinarily large runtime despite using a data set of moderate size, a runtime that is dependant on the combination size used during the search.

3.3 Detection Power

Using the simulated data, the detection power of the different methods can be measured as the ratio of data sets for which the epistatic interaction is correctly identified. Two different detection power metrics were used in the evaluation: the detection power considering all interactions reported by each method, and the detection power when only the first interaction reported is considered. Some implementations provide its output as a list of combinations in no particular order, therefore only the detection power of all reported interactions is obtainable. These methods include BADTrees, StepPLR, MACOED, NHSA-DHSC, ATHENA and BHIT. On the other side, some methods only report a single interaction,

thus both detection powers will be identical. These methods are MDR, LRWM, GALE and BEAM3.

All the programs were executed using a total of 192 CPU cores from the clusters described in Supplementary Table S1, available online. Since programs are executed repeatedly using different data, program-level parallelism can be exploited by running multiple instances of the same program using different CPU cores. All the results from each of the programs shown during the evaluation could be obtained within a week's time. MDR, EpiMiner and CINOEDV were excluded from fourth-order evaluation due to their unreasonably high runtimes. HiSeeker was also excluded from the fourth-order evaluation due to errors during execution.

Given the large number of configurations used, it is impractical to present all the individual results. Therefore, in this evaluation, the results are grouped by the interaction order and by the type of epistatic relationships, since these two account for most of the variation between results from the same method. The complete results are available in Tables S2, S3 and S4 of the Supplementary material, available online.

Epistasis With Marginal Effects Following an Additive Model

Fig. 2 shows the detection power of all methods when the data contains epistatic interactions displaying marginal effects under the additive interaction model. The first subfigure represents the detection power from each method when all the reported interactions are considered, and the second subfigure represents the same detection power when only the first reported interaction is considered.

Exhaustive methods reliably find the epistatic interaction in virtually all cases, and the correct interaction is the one always reported first. Conversely, genetic algorithms almost always miss the epistatic interaction. The remainder of the methods show mixed results and have to be discussed individually.

Out of the filtering methods, EDCF and SingleMI perform best with maximum detection powers even when considering only the first reported interaction. MECPM follows closely, although its detection power takes a toll when increasing the interaction order or when only the first reported interaction is considered. LAMPLINK and EpiMiner's success can only be seen in third-order interactions when all of the reported are considered, DCHE shows mediocre results, and Mendel and Hiseeker cannot locate interactions whatsoever.

Depth-first methods show polarizing results. On the one hand, FDHE-IW perfectly identifies the correct interaction. BADTrees also shows a good detection power, although its output includes noise SNPs that do not contribute to the phenotypic outcome. LRMW, StepPLR and SNPRuler, on the other hand, obtain very low (if not zero) detection powers.

Swarm intelligent methods show quite different results attending to the order of the interaction, with the only exception of IACO. This is coherent with the parameterization employed, since the number of iterations and agents (which control how much of the search space is explored) is kept constant throughout the evaluation despite the search space growing when the interaction order is increased. Swarm intelligent methods are also the most affected ones when only the first interaction is considered. IACO obtains almost perfect detection powers when all reported interactions are considered, however its detection power significantly drops when only the first one is used. epiACO and NHSA-DHSC also obtain high detection powers for third-order interactions, but their performance drops significantly when moving to fourth-order. EACO obtains mediocre results for third order, which also drop for fourth-order, and MACOED, AntMiner and CINOEDV obtain poor results.

Lastly, random-search based methods also obtain mixed results. SNPHarvester reports the correct interaction as the first one in almost all data sets. BEAM3 obtains relatively good results, and BHIT is not capable of finding interactions.

Epistasis With Marginal Effects Following a Threshold Model

Fig. 3 shows the detection power of all methods when the data contains epistatic interactions displaying marginal effects under the threshold interaction model. The two subfigures represent the detection power when all interactions or only the first reported are considered, respectively.

Results for the threshold epistatic model are remarkably similar to those of the additive epistatic model, with some minor differences. Exhaustive methods noticeably drop their detection power, while genetic algorithms again fail to find any epistatic interaction.

Out of the filtering methods, HiSeeker DCHE and LAMPLINK present the most drastic changes. HiSeeker goes from not being able to detect interactions at all under the additive epistatic model to reporting the correct interaction as the first one in almost all cases, and DCHE approximately doubles its previous detection power. LAMPLINK, on the contrary, drops its detection power down to zero. EpiMiner and EDCF slightly drop their detection powers. SingleMI and Mendel obtain very similar results compared to previous additive model results, the former with high powers and the later with powers next to zero.

Depth-first methods obtain similar results compared to their previous values, with the only exception of StepPLR. FDHE-IW and BADTrees obtain almost the same detection powers as with the additive model, while LRMW slightly improves it. StepPLR, on the contrary, increases its detection power from next to zero to next to one.

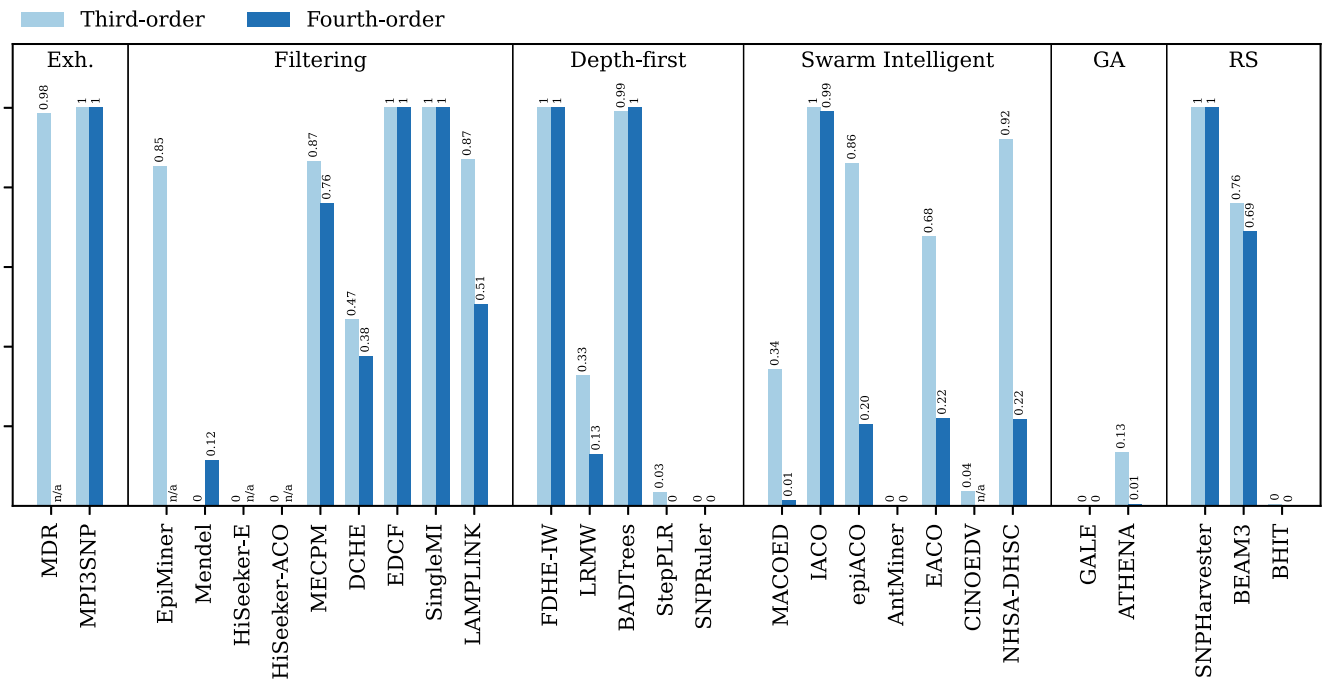
Swarm intelligent algorithms show slight variations from their previous detection powers, with epiACO, AntMiner, CINOEDV and NHSA-DHSC showing similar results while EACO significantly increasing its detection power and IACO and MACOED showing a noticeable decrease.

Random-search based algorithms also show minor variations compared to the results with the additive model. SNPHarvester noticeably drops its detection power for fourth-order interactions, both when all and only the first reported interactions are considered, while maintaining its third-order power. BEAM3, on the opposite, increases its detection power, and BHIT remains near zero.

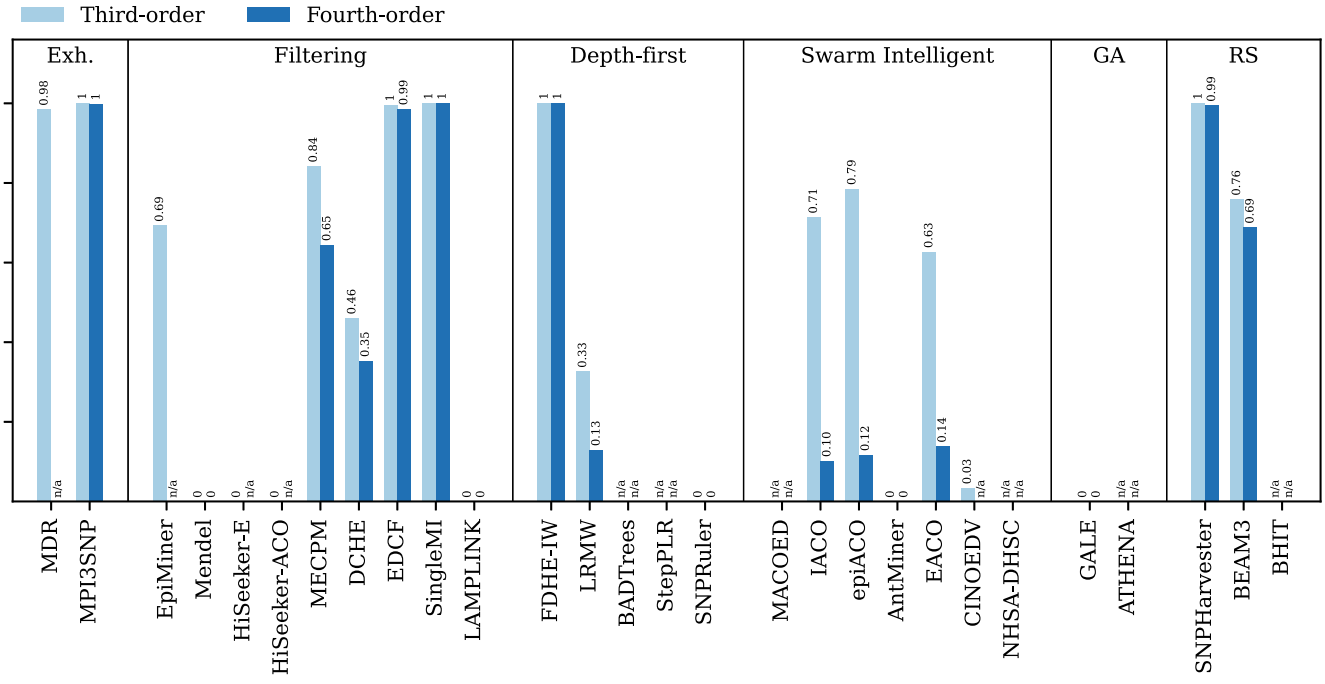
Epistasis With no Marginal Effects Under no Interaction Model

Fig. 4 shows the detection power of all methods when the data contains epistatic interactions displaying no marginal effects under no interaction model.

Detection powers when no marginal effects are present show a completely different story than the previous two interaction models. Out of all the methods tested, only exhaustive approaches are capable of consistently locating



(a) Detection power when all reported interactions are considered.



(b) Detection power when only the first reported interaction is considered.

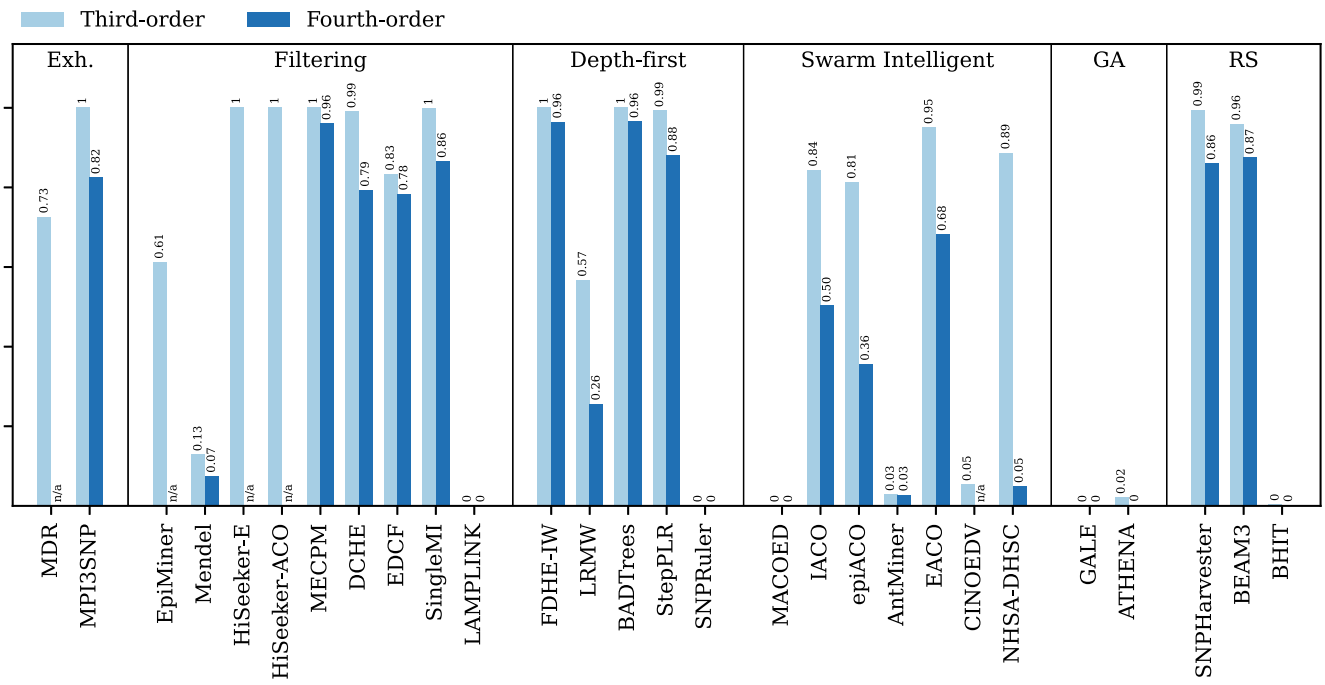
Fig. 2. Detection power of all methods, considering all (a) and only the first (b) reported interactions, for data sets containing epistasis with marginal effects following an additive interaction model. Results not available are labeled accordingly.

interactions that show no marginal effects. The only other methods that show a detection power above zero for third-order interactions are DCHE, EDCF and SNPRuler. DCHE and EDCF show a detection power much lower than in scenarios with marginal effects. SNPRuler, however, was unable to find any interaction in previous interaction models and now it is one of the three methods capable of finding the interaction in a fraction of all data sets.

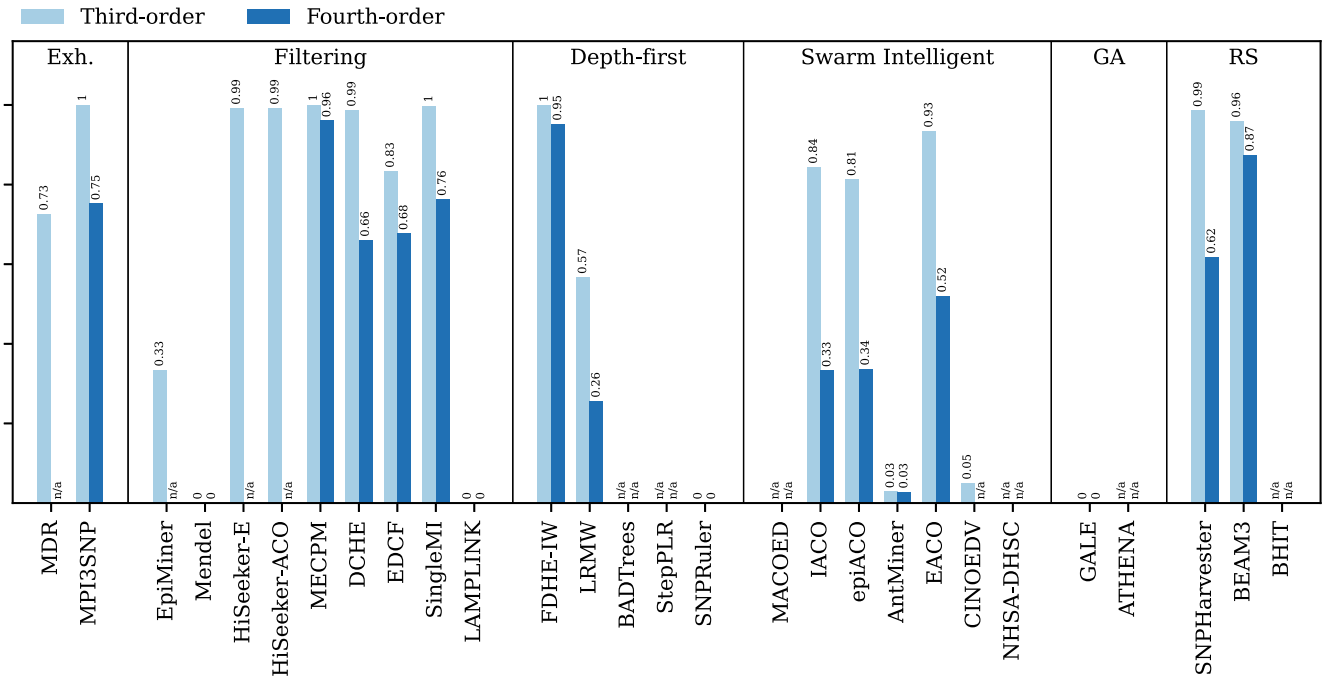
3.4 False Positive Testing

False positive testing evaluates whether or not non-interacting loci are reported when searching for epistasis. To measure false positive detection the Family-Wise Error Rate (FWER) was used, defined as the ratio of data sets where any combination of non-interacting SNPs is reported.

FWER was measured using the previously presented data sets that contain epistatic interactions showing marginal



(a) Detection power when all reported interactions are considered.

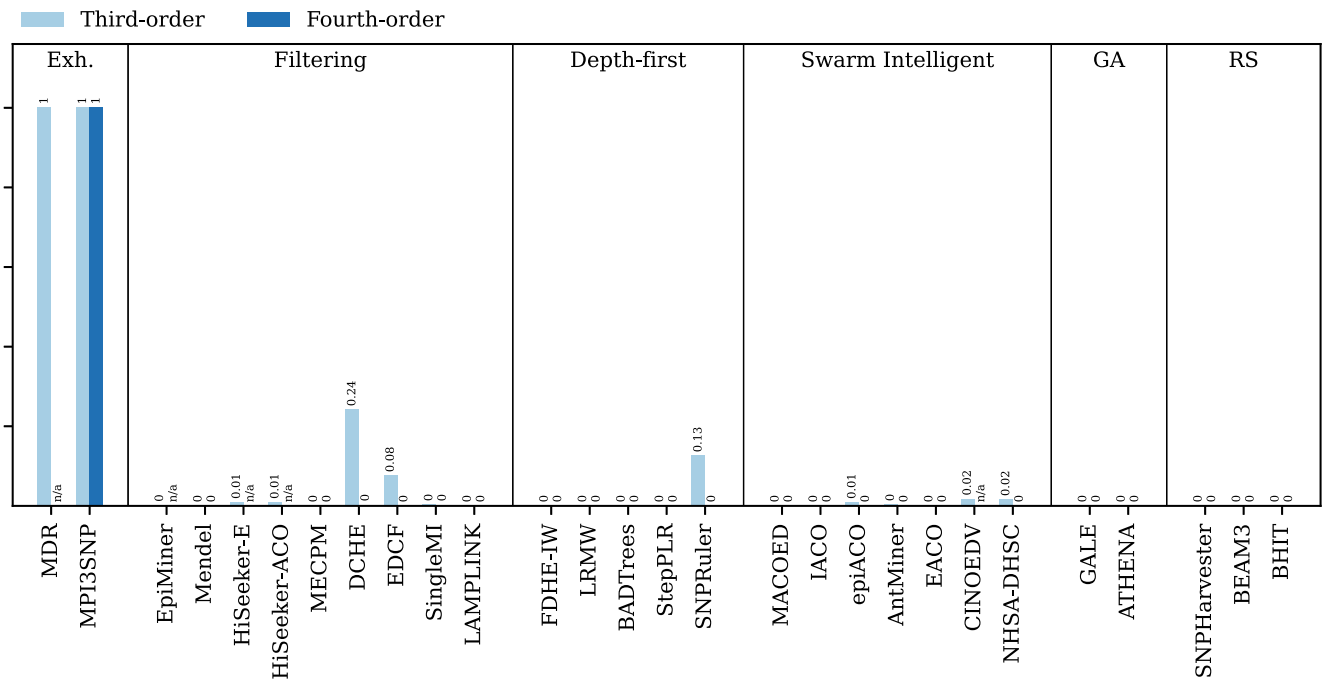


(b) Detection power when only the first reported interaction is considered.

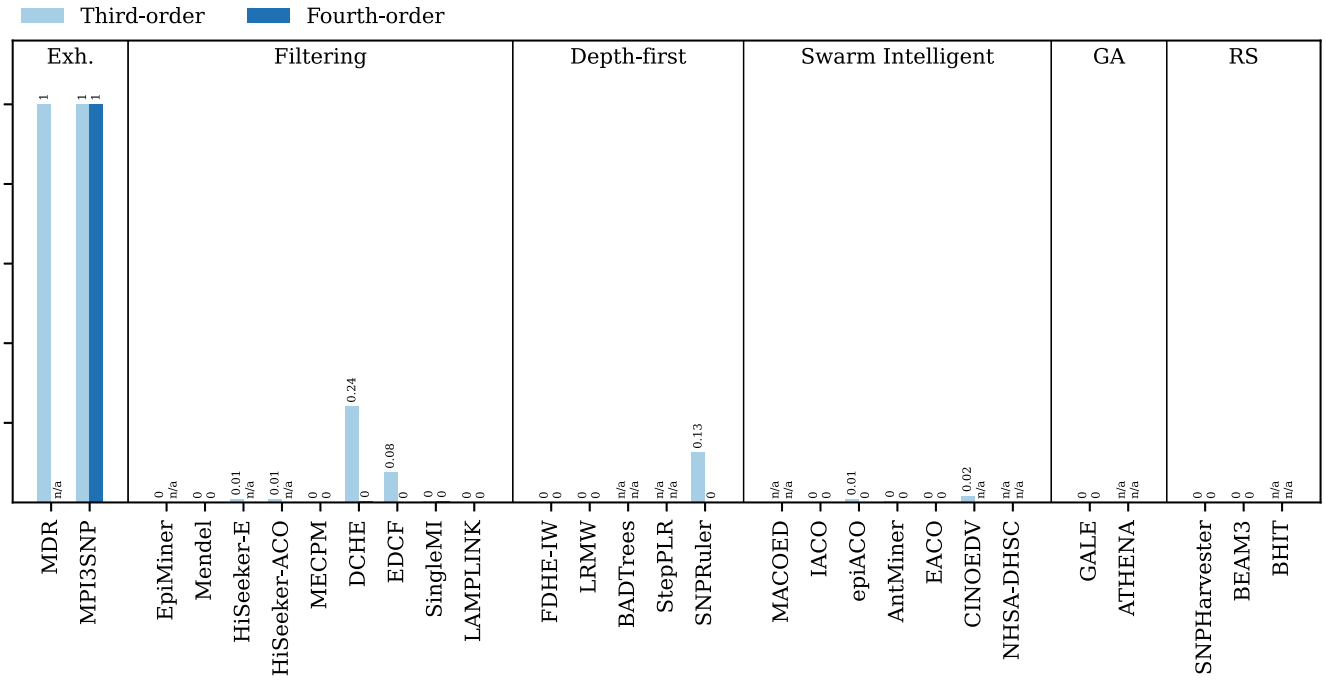
Fig. 3. Detection power of all methods, considering all (a) and only the first (b) reported interactions, for data sets containing epistasis with marginal effects following a threshold interaction model. Results not available are labelled accordingly.

effects following additive and threshold models, as well as those showing no marginal effects under no model assumption. Additionally, FWER was also measured on data sets containing no epistatic interactions.

FWER could not be measured for all epistasis detection methods and for all scenarios presented. Implementations that are forced to return any number of unordered SNP combinations could not be included in this evaluation. This includes LRMW, BADTrees, StepPLR and ATHENA. The FWER for programs that return a fixed number of ordered combinations was measured considering only the first reported interaction. In this scenario, the FWER is the complementary measure of the detection power when only the first reported interaction is considered, and cannot be measured when there is no epistasis. This includes MDR, MPI3SNP, MECPM, SingleMI and CINOEDV.



(a) Detection power when all reported interactions are considered.



(b) Detection power when only the first reported interaction is considered.

Fig. 4. Detection power of all methods, considering all (a) and only the first (b) reported interactions, for data sets containing epistasis with no marginal effects and under no interaction model. Results not available are labelled accordingly.

Fig. 5 represents the FWER for the methods evaluated. The figure shows that false positives have a significant presence in most of the methods. These results can be divided into three categories: methods that report a large number of false positives regardless of the data, methods that report a small number of false positives and methods that show very different results depending on the epistasis model or presence/absence of epistasis.

Most of the methods fall under the first category. EpiMiner, Mendel, HiSeeker, EDCF, IACO, epiACO, AntMiner, EACO, CINOEDV, NHSA-DHSC and GALE almost always include false positives in its output. On the opposite, MDR, MPI3SNP, SNPRuler, MACOED and BEAM3 keep their FWER under control.

As for methods showing different results depending on the dataset, the most common behaviour is to report false

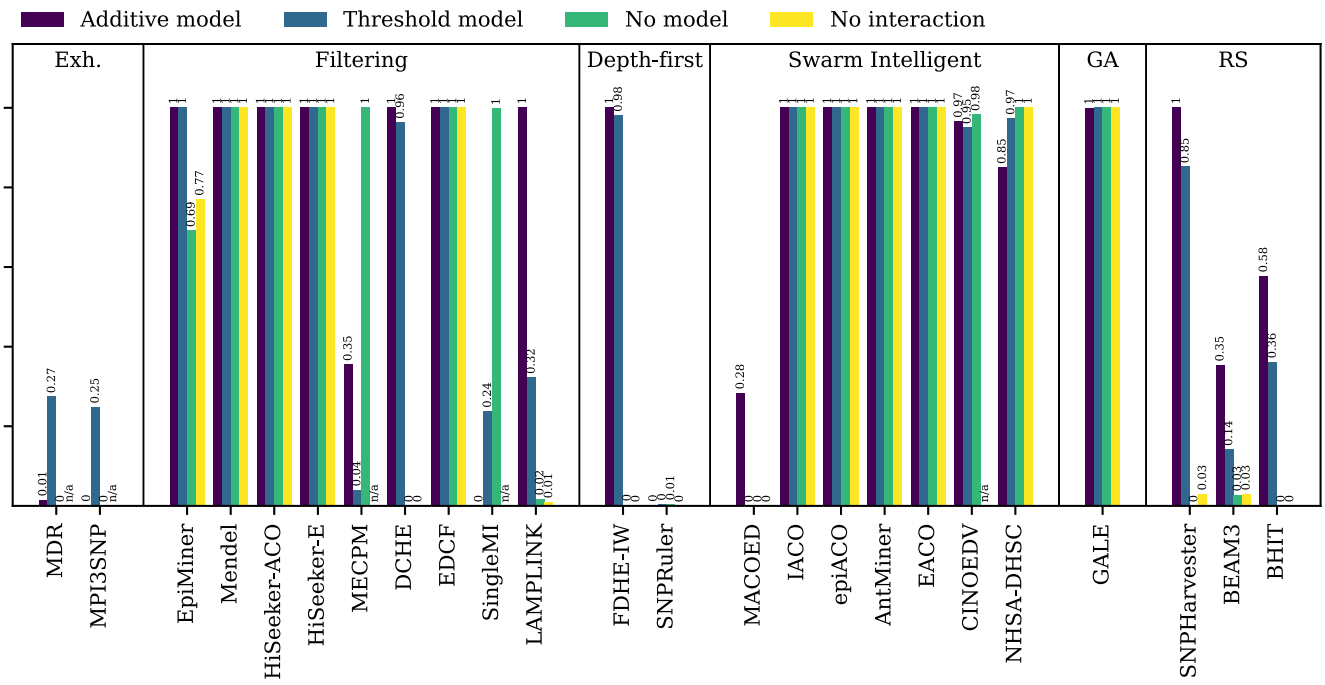


Fig. 5. FWER of all applicable methods, using data sets containing interactions showing marginal effects and following an additive and threshold model, containing interactions showing no marginal effects and under no epistasis model, and containing no interactions.

positives on the presence of marginal effects. DCHE, LAMPLINK, SNPHarvester, BEAM3 and BHIT report almost no false positives when there are no marginal effects or there is no interaction. On the other hand, MECPM and SingleMI show an erratic behavior of the FWER for different data sets.

4 Discussion

It is clear from the previous detection power results that current epistasis detection methods, outside of the exhaustive approach, rely on the existence of marginal effects to locate the epistatic interaction. The best non-exhaustive approach for interactions showing no marginal effects is DCHE, with a detection power of 24.14 percent for third-order interactions which completely disappears when the order is increased.

Table 3 summarizes the results for epistatic interactions with marginal effects. For each program the average detection power is calculated, differentiating between third and fourth-order. FDHE-IW, MPI3SNP, SingleMI, SNPHarvester, BADTrees, MECPM, EDCF and BEAM3 show average detection powers above 80 percent, both for third and fourth-order epistasis search. IACO, NHSA-DHSC, epiACO and EACO despite showing detection powers above 80 percent for third-order searches, immediately drop by more than 20 points when moving to fourth-order. MDR, on the other hand, cannot obtain fourth-order results in a reasonable runtime, and therefore its success is also limited to third-order.

Genetic algorithms are the only family of methods that is not represented on the upper half of the table. Swarm intelligent methods, despite their mediocre results for fourth-order searches, demonstrate good results for third-order, indicating that the number of agents and iterations selected has to take the order of the interactions into consideration. Genetic algorithms, on the other hand, do not find any success under any of the conditions presented.

Table 4 synthesizes the results for false positive testing, showing the average FWER while differentiating between the presence or absence of epistasis. The table shows that, when looking for epistasis, only five methods report false

TABLE 3
Average Detection Power Results for Third and Fourth Order Epistatic Interactions With Marginal Effects

Rank	Third-order		Fourth order	
	Method	Power	Method	Power
1	FDHE-IW	100.00 %	BADTrees	97.90 %
2	MPI3SNP	100.00 %	FDHE-IW	97.85 %
3	SingleMI	99.88 %	SingleMI	91.90 %
4	SNPHarvester	99.71 %	SNPHarvester	91.60 %
5	BADTrees	99.50 %	MPI3SNP	89.45 %
6	MECPM	93.29 %	MECPM	88.00 %
7	IACO	92.17 %	EDCF	87.00 %
8	EDCF	91.67 %	BEAM3	80.00 %
9	NHSA-DHSC	90.38 %	IACO	69.75 %
10	BEAM3	85.92 %	DCHE	62.60 %
11	MDR	85.54 %	StepPLR	52.80 %
12	epiACO	83.67 %	EACO	49.70 %
13	EACO	81.29 %	epiACO	29.55 %
14	EpiMiner	73.25 %	LRMW	20.55 %
15	DCHE	72.88 %	LAMPLINK	20.25 %
16	StepPLR	51.33 %	NHSA-DHSC	11.60 %
17	HiSeeker-ACO	50.00 %	Mendel	9.05 %
18	HiSeeker-E	50.00 %	AntMiner	1.65 %
19	LRMW	44.67 %	MACOED	0.55 %
20	LAMPLINK	43.50 %	ATHENA	0.30 %
21	MACOED	17.13 %	BHIT	0.05 %
22	ATHENA	7.75 %	GALE	0.00 %
23	Mendel	6.46 %	SNPRuler	0.00 %
24	CINOEDV	4.46 %	CINOEDV	-
25	AntMiner	1.42 %	EpiMiner	-
26	BHIT	0.38 %	HiSeeker-ACO	-
27	GALE	0.00 %	HiSeeker-E	-
28	SNPRuler	0.00 %	MDR	-

TABLE 4
Average FWER Results When Epistasis is Present and Absent

Rank	With epistasis		Without epistasis	
	Method	FWER	Method	FWER
1	SNPRuler	0.29%	BHIT	0.00%
2	MPI3SNP	5.44%	DCHE	0.00%
3	MDR	11.19%	FDHE-IW	0.00%
4	BEAM3	16.11%	MACOED	0.00%
5	MACOED	18.84%	SNPRuler	0.00%
6	SingleMI	25.18%	LAMPLINK	1.00%
7	MECPM	29.47%	BEAM3	3.00%
8	BHIT	35.60%	SNPHarvester	3.00%
9	LAMPLINK	51.98%	EpiMiner	77.00%
10	SNPHarvester	76.62%	AntMiner	100.00%
11	DCHE	79.18%	EACO	100.00%
12	FDHE-IW	79.98%	EDCF	100.00%
13	NHSA-DHSC	87.42%	epiACO	100.00%
14	EpiMiner	93.03%	GALE	100.00%
15	CINOEDV	96.35%	HiSeeker-ACO	100.00%
16	GALE	99.98%	HiSeeker-E	100.00%
17	AntMiner	100.00%	IACO	100.00%
18	EACO	100.00%	Mendel	100.00%
19	EDCF	100.00%	NHSA-DHSC	100.00%
20	epiACO	100.00%	CINOEDV	-
21	HiSeeker-ACO	100.00%	MDR	-
22	HiSeeker-E	100.00%	MECPM	-
23	IACO	100.00%	MPI3SNP	-
24	Mendel	100.00%	SingleMI	-

positives in less than 25 percent of the data sets tested. These methods are SNPRuler, MPI3SNP, MDR, BEAM3 and MACOED. Only three of these five methods show good detection powers, which questions if the good false positive results of SNPRuler and MACOED are linked to their lack of detection.

When epistasis is not present, eight methods can obtain FWER close to zero. Out of these eight, half obtains reasonably high detection powers when epistasis is present, including DCHE, FDHE-IW, BEAM3 and SNPHarvester. The other half, composed of BHIT, MACOED, SNPRuler and LAMPLINK, obtains poor detection powers which, again, questions if the good false positive results are linked to their weak detection ability.

Results also suggest a possible two-stage strategy for finding new epistatic interactions with marginal effects, in a reasonable execution time and with a low probability of including false positives: combining FDHE-IW with MPI3SNP. FDHE-IW could be used first to discern whether or not a data set contains epistasis, due to its high detection power, low runtime and low FWER under the assumption of no epistasis. If any candidate combination is reported, MPI3SNP would then be used to analyse only the reported SNPs due to its high detection power and low FWER, under the assumption of epistasis, while circumventing the high runtime associated with exhaustive methods due to the previous filtering step.

To conclude the evaluation, it is worth mentioning that BADTrees, a method that achieves very good results in terms of detection power, does not implement any statistical method that allows the elimination of false positives, which detracts from the tool's applicability.

5 Conclusion

Epistasis detection is an area of GWAS under active research. High-order epistasis has been speculated to be the source of complex traits, however there is no extensive study that empirically compares the state-of-the-art methods in this regard. This work provides an overview of the current methods dedicated to high-order epistasis detection, as well as a comparison of the results achieved by the different implementations in terms of detection power and type I error rate.

Results show that many of the current epistasis detection methods, regardless of the strategy used, can reliably find the epistatic interaction when marginal effects are present, although their detection power generally decreases with the order of the interaction. The only exception are genetic algorithms, as none of the two methods implementing this strategy can consistently find interactions.

Non-exhaustive methods, however, behave very poorly when marginal effects are absent. In this scenario the only option that seems to reliably locate the interactions is the exhaustive strategy, with the subsequent exponential runtime complexity associated with the order of the interaction searched.

False positives evaluation speaks of a different story. Out of the 27 methods compared, BEAM3 is the only method capable of reliably finding epistasis while keeping type I errors to a minimum. Moving forward, authors should give more importance to type I error control. Methods that consistently report false positives lose much of their value, since their usability is restricted to the verification of previous findings. Looking for new epistatic interactions requires implementing a tight false positive control in order to avoid reporting false associations.

Outside of the results, there are other considerations to make out of this study:

- The difficulty of appropriately using the programs. Most of the programs require user input to select a number of configuration parameters. These parameters can have a direct impact on the detection power of the tool, as made evident by ACO methods. Despite this, most of the programs have insufficient documentation on what each parameter does or how to select them. Authors should either pay more attention to the documentation so that a better-informed decision can be made or avoid leaving the choice to the user by automatically selecting these parameters based on the problem size or previous results.
- The lack of standardization in the input format used by the different tools. Each author arbitrarily decides the format used in his/her program, with no regards towards integrability with other software tools or ease of use for the end-user. We would also like to see more standardization in the whole process for these types of studies. Agreeing on a format to use would facilitate the incorporation of newly developed software in existing pipelines for analysis of genotype data, without the need of adding layers of scripts to transform one format into another or interpret the results differently depending on the program used.

- The lack of agreement on how to evaluate the performance of the tools. Each author, in his own work, either develops an ad-hoc benchmarking data set to compare his new program with some other epistasis detection tools, or reuses the data from the simulation study of some other comparison. This evaluation methodology makes the contrast of different epistasis studies difficult since the simulation conditions are mostly different. Developing a common benchmark of data sets to employ during the evaluation would allow for the comparison of all published epistasis tools without the need for repeating the analysis in each of the evaluations.

Acknowledgments

This research was funded by the Ministry of Economy and Competitiveness of Spain (CGL2016-75482-P), Ministry of Science and Innovation of Spain (TIN2016-75845-P and PID2019-104184RB-I00, AEI/FEDER/EU, 10.13039/501100011033), the Xunta de Galicia (Grupo de Referencia Competitiva, ED431C2016-037), the Xunta de Galicia and FEDER funds of the EU (Centro de Investigación de Galicia accreditation 2019-2022, ref. ED431G2019/01), Consolidation Program of Competitive Reference Groups (ED431C 2017/04) and the FPU Program of the Ministry of Education of Spain (FPU16/01333). The authors would like to thank the Supercomputing Center of Galicia (CESGA) for providing access to its supercomputing facilities and assisting with the problems encountered, as well as all of the authors contacted for assistance while using their epistasis detection applications.

References

[1] E. López-Cortegano and A. Caballero, "Inferring the nature of missing heritability in human traits using data from the GWAS catalog," *Genetics*, vol. 212, no. 3, pp. 891–904, 2019.

[2] R. Rieger, A. Michaelis, and M. M. Green, *Glossary of Genetics and Cytogenetics: Classical and Molecular*. Berlin, Germany: Springer, 2012.

[3] J. Domingo, P. Baeza-Centurion, and B. Lehner, "The causes and consequences of genetic interactions (epistasis)," *Annu. Rev. Genomics Hum. Genet.*, vol. 20, pp. 433–460, 2019.

[4] A. Carvajal-Rodriguez, K. A. Crandall, and D. Posada, "Recombination favors the evolution of drug resistance in HIV-1 during antiretroviral therapy," *Inf. Genet. Evol.*, vol. 7, no. 4, pp. 476–483, 2007.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2009.

[6] M. B. Taylor and I. M. Ehrenreich, "Higher-order genetic interactions and their contribution to complex traits," *Trends Genet.*, vol. 31, no. 1, pp. 34–40, 2015.

[7] Z. R. Sailer and M. J. Harms, "High-order epistasis shapes evolutionary trajectories," *PLoS Comput. Biol.*, vol. 13, no. 5, 2017, Art. no. e1005541.

[8] A. Sanchez-Gorostiaga, D. Bajić, M. L. Osborne, J. F. Poyatos, and A. Sanchez, "High-order interactions distort the functional landscape of microbial consortia," *PLoS Biol.*, vol. 17, no. 12, 2019, Art. no. e3000550.

[9] J. Shang *et al.*, "A review of ant colony optimization based methods for detecting epistatic interactions," *IEEE Access*, vol. 7, pp. 13 497–13 509, 2019.

[10] S. Uppu, A. Krishna, and R. P. Gopalan, "A review on methods for detecting SNP interactions in high-dimensional genomic data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 2, pp. 599–612, Mar./Apr. 2018.

[11] X. Ding and X. Guo, "A survey of SNP data analysis," *Big Data Mining Anal.*, vol. 1, no. 3, pp. 173–190, 2018.

[12] A. Upton, O. Trelles, J. A. Cornejo-García, and J. R. Perkins, "Review: High-performance computing to detect epistasis in genome scale data sets," *Brief. Bioinf.*, vol. 17, no. 3, pp. 368–379, 2016.

[13] C. Niel, C. Sinoquet, C. Dina, and G. Rocheleau, "A survey about methods dedicated to epistasis detection," *Front. Genet.*, vol. 6, 2015, Art. no. 285.

[14] W.-H. Wei, G. Hemani, and C. S. Haley, "Detecting epistasis in human complex traits," *Nat. Rev. Genet.*, vol. 15, no. 11, pp. 722–733, 2014.

[15] C. L. Koo, M. J. Liew, M. S. Mohamad, and A. H. Mohamed Salleh, "A review for detecting gene-gene interactions using machine learning methods in genetic epidemiology," *BioMed Res. Int.*, vol. 2013, p. 13, 2013, Art no. 432375. [Online]. Available: https://doi.org/10.1155/2013/432375

[16] S. Tuo, H. Chen, and H. Liu, "A survey on swarm intelligence search methods dedicated to detection of high-order SNP interactions," *IEEE Access*, vol. 7, pp. 162 229–162 244, 2019.

[17] C. Chatelain, G. Durand, V. Thuillier, and F. Augé, "Performance of epistasis detection methods in semi-simulated GWAS," *BMC Bioinf.*, vol. 19, no. 1, 2018, Art. no. 231.

[18] J. Shang, J. Zhang, Y. Sun, D. Liu, D. Ye, and Y. Yin, "Performance analysis of novel methods for detecting epistasis," *BMC Bioinf.*, vol. 12, no. 1, 2011, Art. no. 475.

[19] Y. Wang, G. Liu, M. Feng, and L. Wong, "An empirical comparison of several recent epistatic interaction detection methods," *Bioinformatics*, vol. 27, no. 21, pp. 2936–2943, 2011.

[20] C. C. M. Chen, H. Schwender, J. Keith, R. Nunkesser, K. Mengersen, and P. Macrossan, "Methods for identifying SNP interactions: A review on variations of logic regression, random forest and Bayesian logistic regression," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 6, pp. 1580–1591, Nov./Dec. 2011.

[21] M. D. Ritchie and K. V. Steen, "The search for gene-gene interactions in genome-wide association studies: Challenges in abundance of methods, practical considerations, and biological interpretation," *Ann. Transl. Med.*, vol. 6, no. 8, 2018, Art. no. 157.

[22] M. D. Ritchie, "Finding the epistasis needles in the genome-wide haystack," in *Epistasis: Methods and Protocols*, J. H. Moore and S. M. Williams, Eds. New York, NY, USA: Springer, 2015, pp. 19–33.

[23] X. Sun, Q. Lu, S. Mukherjee, P. K. Crane, R. Elston, and M. D. Ritchie, "Analysis pipeline for the epistasis search – statistical versus biological filtering," *Front. Genet.*, vol. 5, 2014, Art. no. 106.

[24] J. Shang, J. Zhang, X. Lei, Y. Zhang, and B. Chen, "Incorporating heuristic information into ant colony optimization for epistasis detection," *Genes Genomics*, vol. 34, no. 3, pp. 321–327, 2012.

[25] S. D. Turner, S. M. Dudek, and M. D. Ritchie, "ATHENA: A knowledge-based hybrid backpropagation-grammatical evolution neural network algorithm for discovering epistasis among quantitative trait Loci," *BioData Mining*, vol. 3, no. 1, 2010, Art. no. 5.

[26] R. T. Guy, P. Santago, and C. D. Langefeld, "Bootstrap aggregating of alternating decision trees to detect sets of SNPs that associate with disease," *Genet. Epidemiol.*, vol. 36, no. 2, pp. 99–106, 2012.

[27] Y. Zhang, "A novel Bayesian graphical model for genome-wide multi-SNP association mapping," *Genet. Epidemiol.*, vol. 36, no. 1, pp. 36–47, 2012.

[28] J. Wang *et al.*, "A Bayesian model for detection of high-order interactions among genetic variants in genome-wide association studies," *BMC Genomics*, vol. 16, no. 1, 2015, Art. no. 1011.

[29] J. Shang, Y. Sun, J.-X. Liu, J. Xia, J. Zhang, and C.-H. Zheng, "CINOEDV: A co-information based method for detecting and visualizing n-order epistatic interactions," *BMC Bioinf.*, vol. 17, 2016, Art. no. 214.

[30] X. Guo, Y. Meng, N. Yu, and Y. Pan, "Cloud computing for detecting high-order genome-wide epistatic interaction via dynamic clustering," *BMC Bioinf.*, vol. 15, 2014, Art. no. 102.

[31] Y. Sun, X. Wang, J. Shang, J. Liu, C. Zheng, and X. Lei, "Introducing heuristic information into ant colony optimization algorithm for identifying epistasis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 4, pp. 1253–1261, Jul./Aug. 2020.

[32] M. Xie, J. Li, and T. Jiang, "Detecting genome-wide epistases based on the clustering of relatively frequent items," *Bioinformatics*, vol. 28, no. 1, pp. 5–12, 2012.

[33] Y. Sun, J. Shang, J.-X. Liu, S. Li, and C.-H. Zheng, "epiACO - A method for identifying epistasis based on ant Colony optimization algorithm," *BioData Mining*, vol. 10, 2017, Art. no. 23.

[34] J. Shang, J. Zhang, Y. Sun, and Y. Zhang, "EpiMiner: A three-stage co-information based method for detecting and visualizing epistatic interactions," *Digit. Signal Process.*, vol. 24, pp. 1–13, 2014.

[35] S. Tuo, "FDHE-IW: A fast approach for detecting high-order epistasis in genome-wide case-control studies," *Genes*, vol. 9, no. 9, 2018, Art. no. 435.
[36] R. J. Urbanowicz and J. H. Moore, "The application of pittsburgh-style learning classifier systems to address genetic heterogeneity and epistasis in association studies," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2010, pp. 404–413.
[37] J. Liu, G. Yu, Y. Jiang, and J. Wang, "HiSeeker: Detecting high-order SNP interactions based on pairwise SNP combinations," *Genes*, vol. 8, no. 6, 2017, Art. no. 153.
[38] Y. Sun, J. Shang, J. Liu, and S. Li, "An improved ant colony optimization algorithm for the detection of SNP-SNP interactions," in *Proc. Int. Conf. Intell. Comput.*, 2016, pp. 21–32.
[39] A. Terada, R. Yamada, K. Tsuda, and J. Sese, "LAMPLINK: Detection of statistically significant SNP combinations from GWAS data," *Bioinformatics*, vol. 32, no. 22, pp. 3513–3515, 2016.
[40] C. Wei and Q. Lu, "GWGGI: Software for genome-wide gene-gene interaction analysis," *BMC Genet.*, vol. 15, no. 1, 2014, Art. no. 101.
[41] 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*, vol. 31, no. 5, pp. 634–641, 2015.
[42] M. D. Ritchie *et al.*, "Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer," *Amer. J. Hum. Genet.*, vol. 69, no. 1, pp. 138–147, 2001.
[43] D. J. Miller *et al.*, "An algorithm for learning maximum entropy probability models of disease risk that efficiently searches and sparingly encodes multilocus genomic interactions," *Bioinformatics*, vol. 25, no. 19, pp. 2478–2485, 2009.
[44] T. T. Wu, Y. F. Chen, T. Hastie, E. Sobel, and K. Lange, "Genome-wide association analysis by lasso penalized logistic regression," *Bioinformatics*, vol. 25, no. 6, pp. 714–721, 2009.
[45] C. Ponte-Fernández, J. González-Domínguez, and M. J. Martín, "Fast search of third-order epistatic interactions on CPU and GPU clusters," *Int. J. High Perform. Comput. Appl.*, vol. 34, pp. 20–29, 2019.
[46] 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," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 11529.
[47] D. Jünger, C. Hundt, J. G. Domínguez, and B. Schmidt, "Speed and accuracy improvement of higher-order epistasis detection on CUDA-enabled GPUs," *Cluster Comput.*, vol. 20, no. 3, pp. 1899–1908, 2017.
[48] 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*, vol. 25, no. 4, pp. 504–511, 2009.
[49] X. Wan, C. Yang, Q. Yang, H. Xue, N. L. S. Tang, and W. Yu, "Predictive rule inference for epistatic interaction detection in genome-wide association studies," *Bioinformatics*, vol. 26, no. 1, pp. 30–37, 2010.
[50] M. Y. Park and T. Hastie, "Penalized logistic regression for detecting gene interactions," *Biostatistics*, vol. 9, no. 1, pp. 30–50, 2008.
[51] T. Uno, M. Kiyomi, and H. Arimura, "LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," in *Proc. IEEE ICDM Workshop Frequent Itemset Mining Implementations*, 2004, vol. 126.
[52] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Comput.*, vol. 8, no. 2, pp. 239–287, 2009.
[53] J. Kennedy, *Swarm Intelligence*. Boston, MA, USA: Springer, 2006, pp. 187–219.
[54] X.-S. Yang, *Harmony Search as a Metaheuristic Algorithm*. Berlin, Germany: Springer, 2009, pp. 1–14.
[55] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, 1994.
[56] X. Llorà and J. M. Garrell, "Knowledge-independent data mining with fine-grained parallel evolutionary algorithms," in *Proc. 3rd Annu. Conf. Genet. Evol. Comput.*, 2001, pp. 461–468.
[57] M. O'Neill and C. Ryan, "Grammatical evolution," *IEEE Trans. Evol. Comput.*, vol. 5, no. 4, pp. 349–358, Aug. 2001.
[58] R. J. Urbanowicz, J. Kiralis, N. A. Sinnott-Armstrong, T. Heberling, J. M. Fisher, and J. H. Moore, "GAMETES: A fast, direct algorithm for generating pure, strict, epistatic models with random architectures," *BioData Mining*, vol. 5, 2012, Art. no. 16.
[59] C. Ponte-Fernández, J. González-Domínguez, A. Carvajal-Rodríguez, and M. J. Martín, "Toxo: A library for calculating penetrance tables of high-order epistasis models," *BMC Bioinf.*, vol. 21, no. 1, 2020, Art. no. 138.
[60] J. Marchini, P. Donnelly, and L. R. Cardon, "Genome-wide strategies for detecting multiple loci that influence complex diseases," *Nat. Genet.*, vol. 37, no. 4, pp. 413–417, 2005.
[61] J.-H. Park *et al.*, "Distribution of allele frequencies and effect sizes and their interrelationships for common genetic susceptibility variants," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 44, pp. 18 026–18 031, 2011.
[62] R. Walters, C. Laurin, and G. H. Lubke, "An integrated approach to reduce the impact of minor allele frequency and linkage disequilibrium on variable importance measures for genome-wide data," *Bioinformatics*, vol. 28, no. 20, pp. 2615–2623, 2012.
[63] A. Caballero, *Quantitative Genetics*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
[64] T. J. Polderman *et al.*, "Meta-analysis of the heritability of human traits based on fifty years of twin studies," *Nat. Genet.*, vol. 47, no. 7, 2015, Art. no. 702.
[65] S. Harari, "Why we should care about ultra-rare disease," *Eur. Respir. Rev.*, vol. 25, pp. 101–103, 2016.

Christian Ponte-Fernández received the BSc degree in computer engineering and MSc degree in bioinformatics for health sciences from the Universidade da Coruña, Spain, in 2016 and 2018, respectively. He is currently working toward the PhD degree in Information Technology Research, Universidade da Coruña, Spain. His research interests include high performance computing, and parallel and distributed algorithms applied to bioinformatics.

Jorge González-Domínguez received the BSc, MSc, and PhD degrees in computer science from the Universidade da Coruña, Spain, in 2008, 2010, and 2013, respectively. He is currently an assistant teacher in the Computer Architecture Group, Universidade da Coruña, Spain. His main research interests are in the areas of high-performance computing for bioinformatics and PGAS programming languages.

Antonio Carvajal-Rodríguez is a faculty member at the Faculty of Biology in the University of Vigo, Spain. He has training in both biology and computer science. His current research is focused on the study of evolution from an information-theoretic approach. He has been working in the description of non-random mating effects in terms of the information theory and have developed a framework for performing multimodel inference on some direct mating parameters. He also maintains the software Myriads for performing multiple testing corrections.

María J. Martín received the BS, MS, and PhD degrees in physics from the University of Santiago de Compostela, Spain, in 1993, 1994, and 1999, respectively. Since 1997, she has been on the faculty of the Department of Computer Engineering, Universidade da Coruña, Spain, where she is currently the chair of the Department and a full professor of Computer Engineering. Her major research interests include parallel algorithms and applications and fault tolerance for parallel applications.