



Toward the automated analysis of complex diseases in genome-wide association studies using genetic programming

Andrew Sohn
University of Pennsylvania
3700 Hamilton Walk
Philadelphia, PA 19143
ansohn@upenn.edu

Randal S. Olson
University of Pennsylvania
3700 Hamilton Walk
Philadelphia, PA 19143
olsonran@upenn.edu

Jason H. Moore
University of Pennsylvania
3700 Hamilton Walk
Philadelphia, PA 19143
jhmoore@upenn.edu

ABSTRACT

Machine learning has been gaining traction in recent years to meet the demand for tools that can efficiently analyze and make sense of the ever-growing databases of biomedical data in health care systems around the world. However, effectively using machine learning methods requires considerable domain expertise, which can be a barrier of entry for bioinformaticians new to computational data science methods. Therefore, off-the-shelf tools that make machine learning more accessible can prove invaluable for bioinformaticians. To this end, we have developed an open source pipeline optimization tool (TPOT-MDR) that uses genetic programming to automatically design machine learning pipelines for bioinformatics studies. In TPOT-MDR, we implement Multifactor Dimensionality Reduction (MDR) as a feature construction method for modeling higher-order feature interactions, and combine it with a new expert knowledge-guided feature selector for large biomedical data sets. We demonstrate TPOT-MDR's capabilities using a combination of simulated and real world data sets from human genetics and find that TPOT-MDR significantly outperforms modern machine learning methods such as logistic regression and eXtreme Gradient Boosting (XGBoost). We further analyze the best pipeline discovered by TPOT-MDR for a real world problem and highlight TPOT-MDR's ability to produce a high-accuracy solution that is also easily interpretable.

CCS CONCEPTS

•Applied computing → Bioinformatics; Genetics; •Computing methodologies → Machine learning; Supervised learning by classification; •Software and its engineering → Genetic programming; Search-based software engineering;

KEYWORDS

genetics, bioinformatics, automated machine learning, multifactor dimensionality reduction, genetic programming, Python

ACM Reference format:

Andrew Sohn, Randal S. Olson, and Jason H. Moore. 2017. Toward the automated analysis of complex diseases in genome-wide association studies using genetic programming. In *Proceedings of the Genetic and Evolutionary*

Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17), 8 pages.

DOI: 10.1145/3071178.3071212

1 INTRODUCTION

We are currently witnessing the explosive growth of technologies that focus on processing the large amounts of data available in the biomedical sciences. Closely, in parallel, machine learning has been gaining traction in an effort toward analyzing and making sense of said biomedical data. However, effectively using machine learning tools often requires deep knowledge of both machine learning techniques as well as the application domain. For example, to effectively apply machine learning to a genome-wide association study (GWAS) [3, 5], the practitioner must understand the complex trait being studied (e.g., a particular disease such as prostate cancer), the research surrounding the underlying genetics of the trait, as well as the numerous steps in the machine learning process that are necessary for a successful analysis (e.g., feature engineering, model selection, etc.). If we can provide off-the-shelf tools that reduce the barrier to entry for using machine learning by non-experts, then such tools could prove beneficial to researchers working in the biomedical sciences. Mapping statistical inferences and models from genetic data analysis to underlying biological processes is an important goal to the field of computational genomics [23].

In recent years, evolutionary computation (EC) has been proven successful in automating a variety of tasks, and even outperformed several hand-designed solutions in human vs. machine competitions [9, 12, 18, 37]. As such, we believe there is considerable promise in using EC to automate the analysis of biomedical data. Last year, we introduced the Tree-Based Pipeline Optimization Tool (TPOT) [31, 33], which seeks to automate the process of designing machine learning pipelines using genetic programming (GP) [1]. We found that TPOT often outperforms a standard machine learning analysis, all the while requiring no *a priori* knowledge about the problem it is solving [30, 32]. Here, we report on our attempts to specialize TPOT for human genetics research.

Human genetics research poses a unique data analysis challenge due to the effects of non-additive gene-gene interactions (i.e., epistasis) and the large number of genes that must be simultaneously considered as possible predictors of a complex trait [26]. As a result, linear models of complex traits often predict little about the trait, and it is typically impossible to perform an exhaustive combinatorial search of every possible genetic model including two or more genes. For this reason, many researchers leverage *a priori* expert knowledge to intelligently reduce and guide the search space when performing a combinatorial search of possible genetic models [28].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright is held by the owner/author(s).
GECCO '17, July 15–19, 2017, Berlin, Germany
ACM 978-1-4503-4920-8/17/07.
<http://dx.doi.org/10.1145/3071178.3071212>

In this paper we introduce TPOT-MDR, which uses GP to automate the study of complex diseases in GWAS. TPOT-MDR automatically designs sequences of common operations from genetic analysis studies, such as data filtering and Multifactor Dimensionality Reduction (MDR) [16, 24, 25, 29, 35], with the goal of producing a model that best predicts the outcome of a complex trait based solely on their genetics. Furthermore, we enable TPOT-MDR to leverage *a priori* expert knowledge through an Expert Knowledge Filter (EKF), which performs feature selection on the GWAS datasets using information from the expert knowledge source.

To demonstrate TPOT-MDR's capabilities, we compare TPOT-MDR to state-of-the-art machine learning methods on a combination of simulated and real-world GWAS datasets. These datasets are all supervised classification datasets with a focus on human disease as the outcome. We find that TPOT-MDR performs significantly better than the state-of-the-art machine learning methods on the GWAS datasets, especially when it is provided the EKF as an optional feature selector. We further analyze the resulting TPOT-MDR model on a real-world GWAS dataset to highlight the interpretability of TPOT-MDR models, which is a feature that is typically lacking in machine learning models. Finally, we release TPOT-MDR as an open source Python software package to be freely used in human genetics research.

2 RELATED WORK

For automated machine learning (AutoML) in general, approaches have mainly focused on optimizing subsets of a machine learning pipeline [19], which is otherwise known as hyperparameter optimization. One readily accessible approach is grid search, which applies brute force search within a search space of all possible model parameters to find the best model configuration. Relatively recently, randomized search [2] and Bayesian optimization (BO) [36] techniques have entered into the foray and have offered more intelligently derived solutions—by adaptively choosing new configurations to train—to the hyperparameter optimization task.

BO has also been applied to the domain of AutoML [8]. Here, the BO iteratively fits a probabilistic model to a set of hyperparameters of a pipeline and their measured performance, chooses and evaluates the most promising hyperparameter, and updates the probabilistic model. GP approaches to AutoML have also been proposed, such as RECIPE [6] and GTMOEP [42], and the aforementioned TPOT [32]. RECIPE uses a grammar-based GP framework, in which a grammar defines a specification for initial population generation as well as for crossover and mutation operations. GTMOEP is similar in spirit to TPOT, but utilizes signal processing functions for its feature construction and selection methods.

Another approach to AutoML has been proposed in [20]. Here, the author computed several meta-features from existing datasets and tied those meta-features to the performance of numerous ML algorithms on those datasets in a “Meta-Learning Space.” When a new dataset was encountered, the author computed the same meta-features from the dataset and recommended the ML algorithm configurations that worked best on datasets with similar meta-features. This technique was shown to work better than using default models and parameters, but was limited by the existing corpus of datasets from which to meta-learn.

Narrowing the focus to AutoML in bioinformatics, the literature is far more sparse. One such example is [11], in which they analyze metabolomics data using a modified Bayesian optimization algorithm integrated with the classification algorithms provided in WEKA, a suite of machine learning software written in Java. The Bayesian optimization provided feature subset selection, which filtered irrelevant and redundant features from the datasets to achieve dimensionality reduction. These techniques lead to an improvement of classification accuracy.

Genetic programming and evolutionary computation methods have also been successfully applied to bioinformatics studies, such as [27, 38], but they do not focus on designing and tuning a series of standard data analysis operations for a specific dataset. As such, although they are related techniques, they do not fall into the automated machine learning domain. In this work, we seek to specialize TPOT-MDR as an AutoML tool for bioinformatics studies.

3 METHODS

In this section, we briefly review TPOT [30–33] and describe the new pipeline operators that were implemented for TPOT-MDR. Afterwards, we describe the datasets used to evaluate TPOT-MDR and compare it to the state-of-the-art machine learning methods.

3.1 TPOT Review

TPOT uses an evolutionary algorithm to automatically design and optimize a series of standard machine learning operations (i.e., a pipeline) that maximize the final classifier's accuracy on a supervised classification dataset. It achieves this task using a combination of genetic programming (GP) [1] and Pareto optimization (specifically, NSGA2 [7]), which optimizes over the trade-off between the number of operations in the pipeline and the accuracy achieved by the pipeline.

TPOT implements four main types of pipeline operators: (1) preprocessors, (2) decomposition, (3) feature selection, and finally (4) models. All the pipeline operators make use of existing implementations in the Python scikit-learn library [34]. Preprocessors consist of two scaling operators to scale the features and an operator that generates new features via polynomial combinations of numerical features. Decomposition consists of a variant of the principal component analysis (RandomizedPCA). Feature selection implements various strategies that serve to filter down the features by some criteria, such as the linear correlation between the feature and the outcome. Models consist of supervised machine learning models, such as tree-based methods, probabilistic and non-probabilistic models, and k-nearest neighbors.

TPOT combines all the operators described above and assembles machine learning pipelines from them. When a pipeline is evaluated, the entire dataset is passed through the pipeline operations in a sequential manner—scaling the data, performing feature selection, generating predictions from the features, etc.—until the final pipeline operation is reached. Once the dataset has fully traversed the pipeline, the final predictions are used to evaluate the overall classification accuracy of the pipeline. This accuracy score is used as part of the pipeline's fitness criteria in the GP algorithm.

To automatically generate and optimize these machine learning pipelines, TPOT uses a GP algorithm as implemented in DEAP [10],

which is a Python package for evolutionary algorithms. Often, GP algorithms build trees of mathematical functions that seek to optimize toward a specified criteria. In TPOT, GP is used to optimize the number and order of pipeline operators as well as each operator's parameters. TPOT follows a standard GP process for 100 generations: random initialization of the initial population (default population size of 100), evaluation of the population on a supervised classification dataset, selection of the most fit individuals on the Pareto front via NSGA2, and variation through uniform mutation (90% of all individuals per generation) and one-point crossover (5% of all individuals per generation). For more information on the TPOT optimization process, see [30].

3.2 TPOT-MDR

TPOT-MDR is a specialized version of TPOT that focuses on genetic analysis studies. It features two new operators that are commonly used genetic analyses of human disease: (1) Multifactor Dimensionality Reduction (MDR) and (2) an Expert Knowledge Filter (EKF).

MDR is a machine learning method for detecting statistical patterns of epistasis by manipulating the feature space of the dataset to more easily identify interactions within the data [16, 24, 25, 35]. To summarize, MDR is a constructive induction algorithm that combines two or more features to create a single feature that captures the interaction affects among the features. This constructed feature can be fed back into the dataset as a new feature or used as the final prediction on the dataset.

The motivation behind adding the EKF operator was that, often times, *a priori* expert knowledge about a biomedical dataset exists: Perhaps the dataset has been analyzed and annotated in previous studies, a database exists with relevant information about the genes in a dataset, or statistical expert knowledge can be derived from the dataset before the study [26]. This *a priori* expert knowledge can be leveraged to guide the TPOT-MDR search algorithm in deciding what genes to include in the final genetic model.

The EKF operator selects an expert knowledge source from the sources provided and selects the N best features according to the expert knowledge source (where N is constrained to $[1, 5]$). Since the EKF operator is parameterized to select both the expert knowledge source and the number of top features to retain, TPOT-MDR optimizes (1) whether and where in the pipeline to include the EKF and (2) the parameters of the EKF. Multiple EKF operators can be included in a TPOT-MDR pipeline, as shown in Figure 1.

Other than the MDR and EKF operators, the only other operators included in TPOT-MDR are a standard univariate feature selection method (SelectKBest in scikit-learn [34], with an evolvable number of features to retain, N , where N is constrained to $[1, 5]$) and a CombinedDFs operator that combines two feature sets together into a single feature set. These operators can be chained together to form a series of operations acting on a GWAS dataset, as depicted in Figure 1. Except for different operator set, the TPOT-MDR optimization process works the same as the original TPOT algorithm as described in Section 3.1, and was run with a population size of 300 for 300 generations with a per-individual mutation rate of 90% and per-individual crossover rate of 5%.

3.3 Datasets

We performed an analysis of TPOT-MDR on both simulated datasets and a real world GWAS dataset. The simulated datasets were generated using GAMETES [39], an open source software package designed to generate GWAS datasets with pure epistatic interactions between the features. We simulated 16 different datasets with specific properties to test the scalability of TPOT-MDR. The simulated datasets included 10, 100, 1,000, or 5,000 single-nucleotide polymorphism (SNP) features, each with 2 predictive features and the remaining features generated randomly using an allele frequency between 0.05 and 0.5. Further, we generated datasets with heritabilities (i.e., noise) of 0.05, 0.1, 0.2, or 0.4, where lower heritability entails more noise in the dataset. Notably, all of the GAMETES datasets had a sample size of 2,000 to ensure a reasonably large dataset size.

By scaling the GAMETES dataset feature spaces from 10 to 5,000, we sought to evaluate how well TPOT-MDR could handle increasingly large numbers of non-predictive features. Similarly, by simulating increasing amounts of noise in the dataset, we sought to evaluate how much noise TPOT-MDR could handle before it failed to detect and model the predictive features. As such, this simulated benchmark provides a detailed view of the strengths and limitations of TPOT-MDR in the GWAS domain.

To validate TPOT-MDR on a real-world dataset, we used a nationally available genetic dataset of 2,286 men of European descent (488 non-aggressive and 687 aggressive cases, 1,111 controls) collected through the Prostate, Lung, Colon, and Ovarian (PLCO) Cancer Screening Trial, a randomized, well-designed, multi-center investigation sponsored and coordinated by the National Cancer Institute (NCI) and their Cancer Genetic Markers of Susceptibility (CGEMS) program. In this study, we focus on prostate cancer aggressiveness as the endpoint, where the prostate cancer is considered aggressive if it was assigned a Gleason score ≥ 7 and was in tumor stages III/IV. Between 1993 and 2001, the PLCO Trial recruited men ages 55–74 years to evaluate the effect of screening on disease specific mortality, relative to standard care. All participants signed informed consent documents approved by both the NCI and local institutional review boards. Access to clinical and background data collected through examinations and questionnaires was approved for use by the PLCO. Men were included in the current analysis if they had a baseline PSA measurement before October 1, 2003, completed a baseline questionnaire, returned at least one Annual Study Update (ASU), and had available SNP profile data through the CGEMS data portal¹. Prior to this study, the CGEMS dataset was filtered to the 219 SNPs associated with biological pathways relevant to aggressive prostate cancer [22]. We call this dataset the “CGEMS Prostate Cancer GWAS dataset.”

For all experiments, we used four different statistical expert knowledge sources as input to the EKF operator: the ReliefF [21], SURF [15], SURF* [14], and MultiSURF [13] algorithms. These algorithms evaluated the entire dataset prior to the experiments and assigned numerical feature importance scores to each feature, which is an indication of how predictive each feature is of the outcome. These numerical scores were provided to the TPOT-MDR EKF operator, and were used to rank the features when filtering the

¹<http://cgems.cancer.gov>

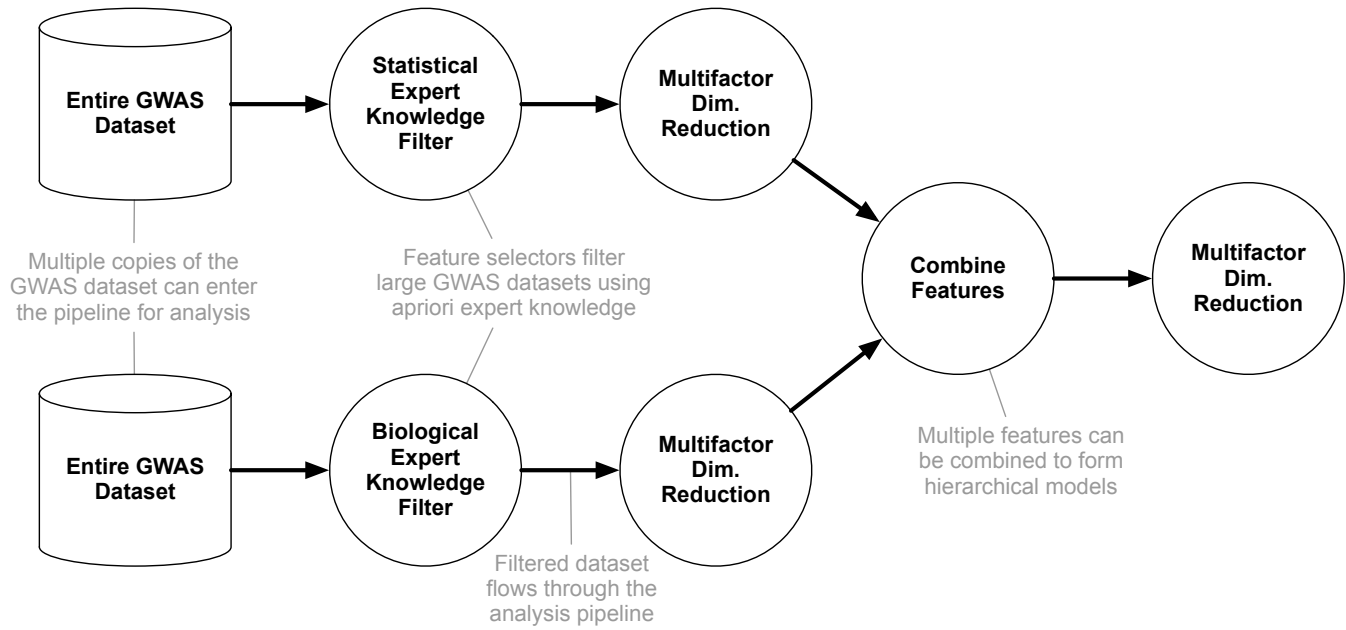


Figure 1: Example TPOT-MDR pipeline. Each circle represents an operation on the dataset, and each arrow represents the passing of the processed dataset to another operation.

datasets. We computed the statistical expert knowledge sources for all 16 GAMETES datasets and the CGEMS Prostate Cancer GWAS dataset, resulting in 68 unique expert knowledge sources (4 for each experiment).

3.4 Evaluating TPOT-MDR

We ran four different sets of experiments on the datasets: (1) Extreme Gradient Boosting (XGBoost)² [4], (2) Logistic Regression³ [17], (3) TPOT-MDR without the EKF, and (4) TPOT-MDR with the EKF. In Section 4, we refer to these experiments as XGBoost, Logistic Regression, TPOT (MDR only), and TPOT (MDR + EKF), respectively. For the GAMETES datasets, we additionally compared the four experiments to the baseline of a MDR model constructed with the two known predictive SNP features (called MDR (Predictive SNPs)), which will achieve the maximum possible classification accuracy for the GAMETES datasets without overfitting on the noisy features.

We chose to compare TPOT-MDR to the XGBoost classifier because XGBoost has been established as a widely popular and successful tree-based classifier in the machine learning community, particularly in the Kaggle⁴ machine learning competitions. Further, we compared TPOT-MDR to a logistic regression to demonstrate the capabilities of a standard linear model on GWAS datasets, which will essentially detect only linear associations between the features and the outcome. Finally, we ran TPOT-MDR without the EKF to demonstrate whether the EKF was important for the TPOT-MDR optimization process.

For every dataset and experiment, we performed 30 replicate runs with unique random number seeds (where applicable). This allowed us to evaluate and explore the limits of TPOT-MDR’s modeling capabilities on a broad range of GWAS datasets, and demonstrate how it performs in comparison to state-of-the-art machine learning methods. In all cases, the accuracy scores reported are averaged balanced accuracy scores from 10-fold cross-validation, where the balanced accuracy metric is a normalized version of accuracy that accounts for class imbalance by calculating accuracy on a per-class basis then averaging the per-class accuracies [40, 41]. With balanced accuracy, a score of 50% is equivalent to random guessing, even with imbalanced datasets.

4 RESULTS

4.1 GAMETES Simulated GWAS Datasets

As shown in Figure 2, TPOT-MDR without the EKF rarely finds the best genetic model because it only has a univariate feature selector at its disposal. In contrast, TPOT-MDR with the EKF always discovers the best genetic model except when there are thousands of features and high noise. Even in the cases where TPOT-MDR with the EKF fails to find the best genetic model, it still discovers better genetic models than the other methods in this study.

For a baseline, we compared TPOT-MDR to a tuned logistic regression and XGBoost, as described in Section 3.4. Figure 2 shows that logistic regression consistently fails to find a good model and barely performs better than chance in even the easiest GAMETES datasets. This finding demonstrates a key flaw in using linear models for GWAS: Linear models will not detect higher-order interactions within the dataset unless the interactions are explicitly modeled. Similarly, XGBoost can sometimes find a good model for

²XGBoost parameters: 500 trees, learning rate 0.0001, and 10 maximum tree depth

³The logistic regression regularization parameter was tuned via 10-fold cross validation

⁴<http://www.kaggle.com>

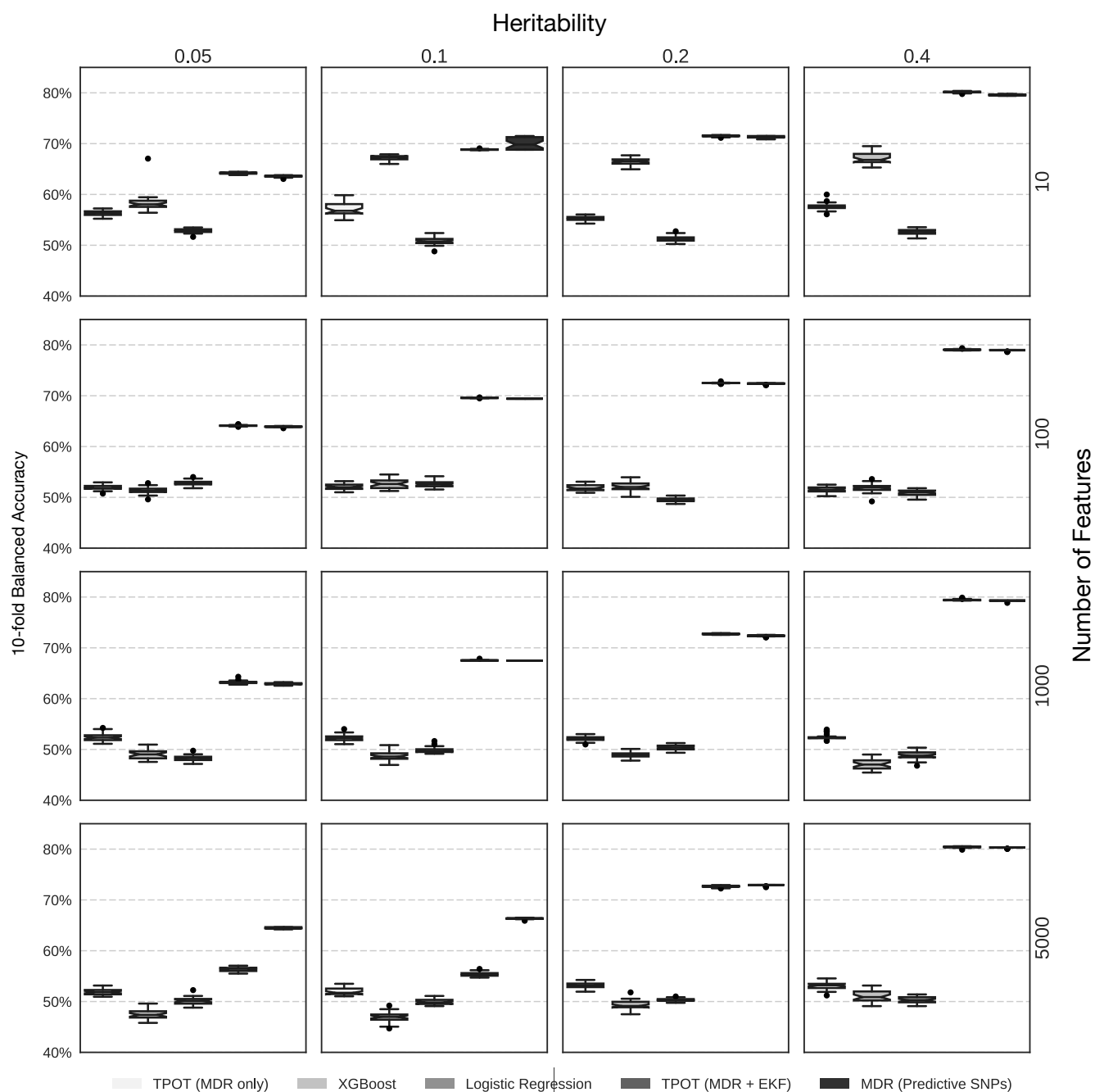


Figure 2: Comparison of results on the simulated GAMETES GWAS datasets. Each box plot shows the distribution of averaged 10-fold balanced accuracies for each experiment, where the notches indicate the 95% confidence interval. A 50% balanced accuracy is equivalent to random guessing. Each panel within the figure corresponds to differing levels of heritability (i.e., dataset noise) and numbers of features in the simulated datasets, ranging from the easiest dataset on the top right (high heritability, small numbers of features) to the hardest dataset bottom left (low heritability, large numbers of features).

Since some of the experiments had little variance in scores, some box plots are too small to determine their color. For clarity, the box plots represent the following experiments, in order from left to right: TPOT (MDR only), XGBoost, Logistic Regression, TPOT (MDR + EKF), and MDR (Predictive SNPs). These experiments are described in Section 3.4.

GWAS datasets if the dataset is heavily filtered beforehand (e.g., to 10s of features), but rapidly degrades in performance as more noisy features are added to the dataset.

4.2 CGEMS Prostate Cancer GWAS Dataset

The CGEMS prostate cancer GWAS dataset has 219 SNPs, 1,175 samples, and likely falls into the “lower heritability” spectrum of the GAMETES datasets. Thus, we would expect to see roughly similar performance on the CGEMS dataset as we saw in the GAMETES datasets with 100 features and 0.1 or 0.05 heritability in Figure 2.

As predicted, Figure 3 shows that XGBoost and logistic regression fail to discover the higher-order interactions within the real-world CGEMS dataset. In contrast, TPOT-MDR with and without the EKF managed to consistently find predictive genetic models for the CGEMS dataset. In particular, TPOT-MDR with the EKF found the best genetic models, largely because the expert knowledge sources (Relieff, SURF, etc.) contained information about the higher-order interactions between the SNPs that TPOT-MDR was able to harness.

To better understand the genetic models that TPOT-MDR discovered, we analyzed the final model from the highest-scoring TPOT-MDR experiment and visualized the pattern of interactions from the MDR model in Figure 4. We see patterns suggestive of statistical epistasis within the model, for example, in the leftmost grid a patient’s aggressive (dark grey cells) or non-aggressive (light grey cells) status can only be determined by a combination of AKT3_rs12031994 and DIABLO_rs12870. Similarly, the pattern of aggressive vs. non-aggressive status between AKT3_rs12031994 and DIABLO_rs12870 varies depending on the state of the third SNP, PRKCQ_rs574512, which suggests a statistical three-way epistatic interaction between the SNPs. If there were no higher-order interactions between the SNPs, then we would expect a patient’s aggressive vs. non-aggressive status to vary independently between the SNPs, i.e., we would expect to see horizontal and vertical bands of aggressive or non-aggressive status within the grids. As previous studies have suggested links between these SNPs and aggressive prostate cancer [22], we can use these TPOT-MDR findings to further elucidate the SNPs’ higher-order interactions and involvement in the development of aggressive prostate cancer in men of European descent.

5 DISCUSSION

In this paper, we introduced a new method and tool, TPOT-MDR, for automating the analysis of complex diseases in genome-wide association studies (GWAS). We developed this tool to aid bioinformaticians so they can more efficiently process and analyze the ever-growing databases of biomedical data. To that end, TPOT-MDR is designed to optimize a series of machine learning operations that are commonly used in biomedical studies, such as filtering the features using expert knowledge sources, combining information from different expert knowledge sources, and modeling the higher-order interactions of the features using Multifactor Dimensionality Reduction (MDR) to predict a patient’s outcome. Before, bioinformaticians would typically perform and refine these operations by hand, whereas now TPOT-MDR can relieve the bioinformatician of these tedious duties so they can focus on more challenging tasks.

Even though this paper focuses on the application of TPOT-MDR to GWAS datasets, we note that TPOT-MDR is a general machine learning tool that will work with any dataset that has categorical features and a binary outcome. TPOT-MDR has been released as a free, open source Python tool and is available on GitHub⁵.

In Section 4, we evaluated TPOT-MDR on a series of simulated and real-world GWAS datasets and found that TPOT-MDR outperforms linear models and XGBoost across all of the datasets (Figures 2 and 3). These findings are important for several reasons. For one, we demonstrated that simple linear models are ill-suited for the analysis of GWAS datasets owing to their inability to model higher-order interactions within the dataset. We also demonstrated that state-of-the-art tree-based machine learning methods—typically thought to be effective at modeling higher-order feature interactions—are similarly ill-suited for modeling GWAS datasets with large numbers of features. Finally, we highlighted the importance of harnessing a *priori* expert knowledge to filter GWAS datasets prior to the modeling step, which could aid state-of-the-art machine learning algorithms such as XGBoost in eliminating extraneous features.

Although the results in Section 4 suggest that TPOT-MDR is superior to the compared methods on every dataset we used, there are some drawbacks to TPOT-MDR that must be considered. For one, linear models and XGBoost are orders of magnitude faster to train and evaluate than TPOT-MDR. As TPOT-MDR uses genetic programming to optimize the series of filtering and modeling operations on the dataset, a single TPOT-MDR run took roughly 3 hours on the CGEMS dataset, whereas XGBoost and logistic regression each took less than a minute. Given that many GWAS datasets often have thousands to hundreds of thousands of SNP features (compared to the 219 in CGEMS), TPOT-MDR will require more work to improve its run time scalability to larger GWAS datasets. Furthermore, TPOT-MDR is highly dependent on its expert knowledge sources. In these experiments, we used expert knowledge sources that specialize in detecting higher-order epistatic interactions, which proved to be critical in both the simulated and real world datasets. If TPOT-MDR is provided with less informative expert knowledge sources, then it will likely perform worse, which we can observe in Figures 2 and 3 (TPOT-MDR without EKF vs. TPOT-MDR with EKF).

As shown in Figure 2, XGBoost can sometimes model higher-order interactions when the dataset is heavily filtered beforehand. However, the resulting XGBoost model is not nearly as interpretable as with TPOT-MDR. TPOT-MDR produces a model that we can inspect to study the pattern of feature interactions within the dataset (Figure 4), whereas XGBoost provides only a complex ensemble of decision trees. This is an important consideration when building machine learning tools for bioinformatics: More often than not, bioinformaticians do not need a black box model that achieves high prediction accuracy on a real-world dataset. Instead, bioinformaticians seek to build a model that can be used as a microscope for understanding the underlying biology of the system they are modeling. In this regard, the models generated by TPOT-MDR can be invaluable for elucidating the higher-order interactions that are often present in complex biological systems.

⁵<https://github.com/rhiever/tpot/tree/tpot-mdr>

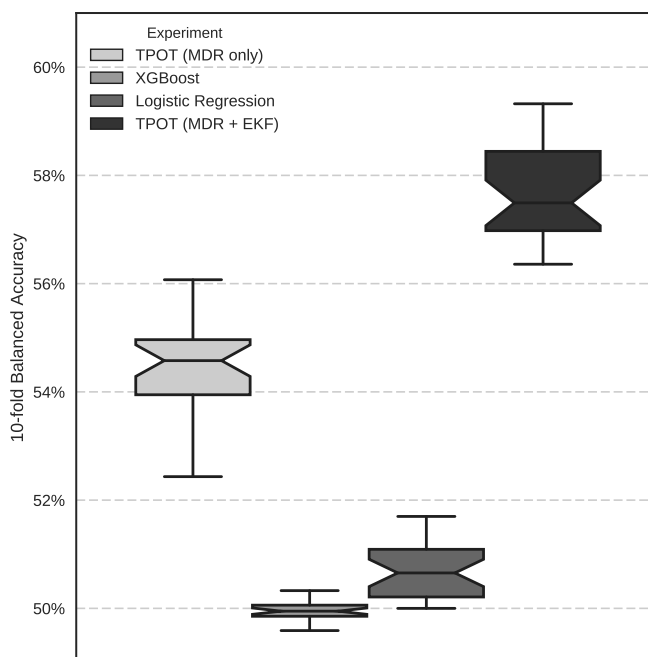


Figure 3: Comparison of results on the CGEMS prostate cancer GWAS dataset. Each box plot shows the distribution of averaged 10-fold balanced accuracies for each experiment, where the notches indicate the 95% confidence interval. A 50% balanced accuracy is equivalent to random guessing.

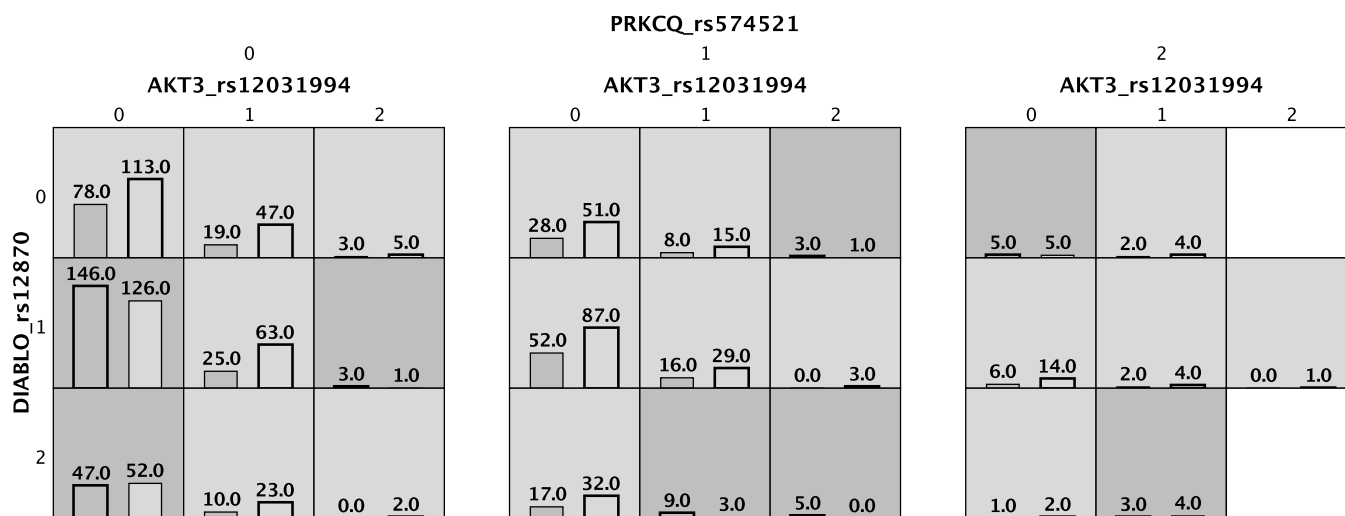


Figure 4: Classification grid for the best MDR model that TPOT-MDR discovered for the CGEMS prostate cancer GWAS dataset. Each of the three grids correspond to one state of the PRKCQ_rs574512 SNP, whereas the cells within each grid correspond to one combination of states between the AKT3_rs12031994 and DIABLO_rs12870 SNPs. Thus, for example, the light grey upper right cell in the leftmost grid corresponds to PRKCQ_rs574512 = 0, AKT3_rs12031994 = 2, and DIABLO_rs12870 = 0.

Dark grey bars and cells indicate aggressive cases (i.e., at risk of aggressive prostate cancer), whereas light grey bars and cells indicate non-aggressive cases (i.e., lower risk of aggressive prostate cancer). The numbers at the top of each bar indicate the number of aggressive and non-aggressive cases that fall within each cell when the entire CGEMS dataset is sorted into the MDR classification grid. If no data points fall into a cell, the cell is left blank.

In conclusion, TPOT-MDR is a promising step forward in using evolutionary algorithms to automate the design of machine learning workflows for bioinformaticians. We believe that evolutionary algorithms (EAs) are poised to excel in the automated machine learning domain, and specialized tools such as TPOT-MDR highlight the strengths of EAs by showing how easily EA solution representations can be adapted to a particular domain.

6 ACKNOWLEDGEMENTS

We thank the Penn Medicine Academic Computing Services for the use of their computing resources. This work was supported by National Institutes of Health grant AI116794.

REFERENCES

- [1] Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. 1998. *Genetic programming: an introduction*. Morgan Kaufmann Publishers San Francisco.
- [2] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *JMLR* 13, Feb (2012), 281–305.
- [3] Adrian Bird. 2007. Perceptions of epigenetics. *Nature* 447, 7143 (2007), 396–398.
- [4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.
- [5] Heather J Cordell. 2009. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics* 10, 6 (2009), 392–404.
- [6] Alex GC de Sá, Walter José GS Pinto, Luiz Otavio VB Oliveira, and Gisele L Pappa. 2017. RECIPE: A Grammar-Based Framework for Automatically Evolving Classification Pipelines. In *European Conference on Genetic Programming*. Springer, 246–261.
- [7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2002), 182–197.
- [8] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*. 2962–2970.
- [9] Stephanie Forrest, ThanhVu Nguyen, Westley Weimer, and Claire Le Goues. 2009. A genetic programming approach to automated software repair. In *GECCO '09*. ACM, 947–954.
- [10] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary algorithms made easy. *JMLR* 13 (2012), 2171–2175.
- [11] Holger Franken, Alexander Seitz, Rainer Lehmann, Hans-Ulrich Häring, Norbert Stefan, and Andreas Zell. 2012. Inferring disease-related metabolite dependencies with a Bayesian optimization algorithm. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 62–73.
- [12] Erik M Fredericks and Betty HC Cheng. 2013. Exploring automated software composition with genetic programming. In *GECCO '13*. ACM, 1733–1734.
- [13] Delaney Granizo-Mackenzie and Jason H Moore. 2013. Multiple threshold spatially uniform relief for the genetic analysis of complex human diseases. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 1–10.
- [14] Casey S Greene, Daniel S Himmelstein, Jeff Kiralis, and Jason H Moore. 2010. The informative extremes: using both nearest and farthest individuals can improve relief algorithms in the domain of human genetics. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 182–193.
- [15] Casey S Greene, Nadia M Penrod, Jeff Kiralis, and Jason H Moore. 2009. Spatially uniform relief (SURF) for computationally-efficient filtering of gene–gene interactions. *BioData Mining* 2, 1 (2009), 5.
- [16] Lance W Hahn, Marylyn D Ritchie, and Jason H Moore. 2003. Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. *Bioinformatics* 19, 3 (2003), 376–382.
- [17] Trevor J Hastie, Robert J Tibshirani, and Jerome H Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, USA.
- [18] Gregory S Hornby, Jason D Lohn, and Derek S Linden. 2011. Computer-automated evolution of an X-band antenna for NASA's Space Technology 5 mission. *Evolutionary Computation* 19 (2011), 1–23.
- [19] Frank Hutter, Jörg Lücke, and Lars Schmidt-Thieme. 2015. Beyond manual tuning of hyperparameters. *KI-Künstliche Intelligenz* 29, 4 (2015), 329–337.
- [20] Alexandros Kalousis. 2002. *Algorithm selection via meta-learning*. Ph.D. Dissertation. Université de Genève.
- [21] Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. 1997. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence* 7, 1 (1997), 39–55.
- [22] Nicole A Lavender, Erica N Rogers, Susan Yeyeodu, James Rudd, Ting Hu, Jie Zhang, Guy N Brock, Kevin S Kimbro, Jason H Moore, David W Hein, and La Creis R Kidd. 2012. Interaction among apoptosis-associated sequence variants and joint effects on aggressive prostate cancer. *BMC Medical Genomics* 5, 1 (2012), 11. DOI: <http://dx.doi.org/10.1186/1755-8794-5-11>
- [23] Chang-Xing Ma, George Casella, and Rongling Wu. 2002. Functional mapping of quantitative trait loci underlying the character process: a theoretical framework. *Genetics* 161, 4 (2002), 1751–1762.
- [24] Jason H Moore and others. 2006. A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. *Journal of Theoretical Biology* 241, 2 (2006), 252–261.
- [25] Jason H Moore and Peter C Andrews. 2015. Epistasis analysis using multifactor dimensionality reduction. *Epistasis: Methods and Protocols* (2015), 301–314.
- [26] Jason H Moore, Folkert W Asselbergs, and Scott M Williams. 2010. Bioinformatics challenges for genome-wide association studies. *Bioinformatics* 26, 4 (2010), 445–455.
- [27] Jason H Moore, Douglas P Hill, Arvis Sulovari, and others. 2013. Genetic analysis of prostate cancer using computational evolution, pareto-optimization and post-processing. In *Genetic Programming Theory and Practice X*. Springer, 87–101.
- [28] Jason H Moore and Bill C White. 2006. Exploiting expert knowledge in genetic programming for genome-wide genetic analysis. In *Parallel Problem Solving from Nature-PPSN IX*. Springer, 969–977.
- [29] Jason H Moore and Scott M Williams. 2002. New strategies for identifying gene–gene interactions in hypertension. *Annals of Medicine* 34, 2 (2002), 88–95.
- [30] Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. 2016. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In *GECCO 2016 (GECCO '16)*. ACM, New York, NY, USA, 485–492. DOI: <http://dx.doi.org/10.1145/2908812.2908918>
- [31] Randal S Olson and Jason H Moore. 2016. Identifying and Harnessing the Building Blocks of Machine Learning Pipelines for Sensible Initialization of a Data Science Automation Tool. arXiv e-print. <http://arxiv.org/abs/1607.08878>. (2016).
- [32] Randal S Olson and Jason H Moore. 2016. TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning. *JMLR* 64 (2016), 66–74.
- [33] Randal S Olson, Ryan J Urbanowicz, Peter C Andrews, Nicole A Lavender, La Creis Kidd, and Jason H Moore. 2016. *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings, Part I*. Springer International Publishing, Chapter Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, 123–137. DOI: http://dx.doi.org/10.1007/978-3-319-31204-0_9
- [34] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. 2011. Scikit-learn: Machine learning in Python. *JMLR* 12 (2011), 2825–2830.
- [35] Marylyn D Ritchie, Lance W Hahn, Nady Roodi, L Renee Bailey, William D Dupont, Fritz F Parl, and Jason H Moore. 2001. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics* 69, 1 (2001), 138–147.
- [36] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *NIPS 2012*. 2951–2959.
- [37] Lee Spector, David M Clark, Ian Lindsay, Bradford Barr, and Jon Klein. 2008. Genetic programming for finite algebras. In *GECCO '08*. ACM, 1291–1298.
- [38] Ryan John Urbanowicz, Angeline S Andrew, Margaret Rita Karagas, and Jason H Moore. 2013. Role of genetic heterogeneity and epistasis in bladder cancer susceptibility and outcome: a learning classifier system approach. *Journal of the American Medical Informatics Association* 20, 4 (2013), 603–612.
- [39] Ryan J Urbanowicz, Jeff Kiralis, Nicholas A Sinnott-Armstrong, Tamra Heberling, Jonathan M Fisher, and Jason H Moore. 2012. GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData Mining* 5 (2012).
- [40] Ryan J Urbanowicz and Jason H Moore. 2015. ExSTraCS 2.0: description and evaluation of a scalable learning classifier system. *Evolutionary Intelligence* 8, 2–3 (2015), 89–116.
- [41] Digna R Velez, Bill C White, Alison A Motsinger, William S Bush, Marylyn D Ritchie, Scott M Williams, and Jason H Moore. 2007. A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. *Genetic Epidemiology* 31 (2007), 306–315. DOI: <http://dx.doi.org/10.1002/gepi.20211>
- [42] Jason Zutty, Daniel Long, Heyward Adams, Gisele Bennett, and Christina Baxter. 2015. Multiple objective vector-based genetic programming using human-derived primitives. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 1127–1134.