GP-Pi: Using Genetic Programming with Penalization and Initialization on Genome-Wide Association Study

Ho-Yin Sze-To[1,⋆], Kwan-Yeung Lee[1], Kai-Yuen Tso[1], Man-Hon Wong[1],
Kin-Hong Lee[1], Nelson L.S. Tang[2,⋆], and Kwong-Sak Leung[1,⋆]

[1] Department of Computer Science and Engineering
{hyszeto,kylee0,kytsomhwong,khlee,ksleung}@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk
[2] Laboratory for Genetics Disease Susceptibility,
Li Ka Shing Institute of Health Sciences
nelsontang@cuhk.edu.hk
http://www.lihs.cuhk.edu.hk
The Chinese University of Hong Kong,
Shatin, N. T., Hong Kong, China

Abstract. The advancement of chip-based technology has enabled the measurement of millions of DNA sequence variations across the human genome. Experiments revealed that high-order, but not individual, interactions of single nucleotide polymorphisms (SNPs) are responsible for complex diseases such as cancer. The challenge of genome-wide association studies (GWASs) is to sift through high-dimensional datasets to find out particular combinations of SNPs that are predictive of these diseases. Genetic Programming (GP) has been widely applied in GWASs. It serves two purposes: attribute selection and/or discriminative modeling. One advantage of discriminative modeling over attribute selection lies in interpretability. However, existing discriminative modeling algorithms do not scale up well with the increase in the SNP dimension. Here, we have developed GP-Pi. We have introduced a penalizing term in the fitness function to penalize trees with common SNPs and an initializer which utilizes expert knowledge to seed the population with good attributes. Experimental results on simulated data suggested that GP-Pi outperforms GPAS with statistically significance. GP-Pi was further evaluated on a real GWAS dataset of Rheumatoid Arthritis, obtained from the North American Rheumatoid Arthritis Consortium. Our results, with potential new discoveries, are found to be consistent with literature.

Keywords: Genome-Wide Association Study, Genetic Programming, Penalization, Initialization, Rheumatoid Arthritis.

1 Introduction

The advancement of chip-based technology has enabled the measurement of millions of DNA sequence variations across the human genome [1, 2]. The by far

[⋆] To whom correspondence should be addressed.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 330–341, 2013.

most common type of such genetic variations are single nucleotide polymorphisms (SNPs), which occur when a different base alternative exists at a single base pair position [3]. In this paper, we focus on studying SNPs, which are categorical variables with 3 outcomes. It is anticipated that at least one SNP occurs approximately every 100 nucleotides across the 3×10^9 nucleotides in human genome [4]. Genome-Wide Association Study (GWAS) is to find out which of the many differences in the genotypes are associated with the phenotypes, or, more specifically, to determine which of the many SNPs are useful for predicting the risk for common diseases, particularly genetic diseases [5]. The role of Computer Science and Bioinformatics is to develop efficient and effective algorithms to identify the disease-associated SNPs. It is a challenging task due to the non-linear mapping between genotypes and phenotypes. SNPs need to be considered jointly in learning algorithms rather than individually. This non-linearity is what we call Epistasis. Here, intelligent algorithms are needed to solve the problem.

1.1 Concept Difficulty

It is believed that high-order interactions of SNPs, not individual SNP, are culprits of complex diseases such as cancer. The challenge of GWAS is to sift through high-dimensional datasets to find out particular combinations of SNPs that are predictive of diseases. Researchers call this a needle-in-a-haystack problem. That is, there may be a particular combination of SNPs which fits well together with a non-linear function and yields good performance when they are used as predictors, e.g. SNP_1 AND SNP_2. However, when these SNPs are considered individually, they may not look different from irrelevant SNPs. Here, the learning algorithm is searching for a genetic haystack, i.e. the number of candidate SNP interactions is tremendous. For J SNPs and pair-wise interaction, J^2 pairs are needed to be considered for epistasis, which is referred to the effect of one locus depending on the genotype of another locus. In general, for J SNPs and K-way interactions, there are O (J^K) candidates [6]. It is computationally infeasible to consider all possible groups of SNP interactions.

1.2 Genetic Programming

Genetic programming (GP) is an automated discovery tool based on Darwinian evolution and natural selection. The goal of GP is to evolve the fittest computer programs to solve problems. Starting from a population of randomly generated computer programs, the GP algorithm evaluates all programs. The good programs are selected, recombined and mutated to form new programs. The process will be terminated after a number of generations or the target fitness is achieved. Genetic programming and its many variations have been applied successfully to a wide range of domains including but not limit to robot vision [7], computational finance [8], drug discovery [9] and motif discovery [10, 11].

1.3 Genetic Programming in GWAS

GP has also been widely applied in GWAS. It serves two purposes: attribute selection [4,12] and discriminative modeling. For attribute selection, studies have demonstrated that GP is a successful wrapper approach in selecting a few important SNPs among thousands of irrelevant ones. It is also found that the use of expert knowledge can significantly improve the performance of GP algorithms [4,12,13].

While the above methods demonstrate satisfactory result in attribute selection, they lack the ability to learn a classification model directly, i.e. discriminative modeling. Discriminative modeling is to learn a classification model to best predict samples susceptibility to disease. While attribute selection may also be performed, the learned model can be directly applied on unseen data to perform classification. One advantage of discriminative modeling approach is that, given the classification model is interpretable, biologists can judge whether the algorithmic output is consistent to biological knowledge and whether it has real-world application value.

GPDTI (Genetic Programming Decision Tree Induction) [14] and GPAS (Genetic Programming for Association Studies) [3] are two discriminative algorithms. They both adopt decision-tree like models to represent their solutions. GPDTI uses basic expression trees and operators to model the problem. GPAS attempts to detect DNFs associated with the response directly by employing multi-valued logic. However, neither GPDTI nor GPAS utilize expert knowledge in guiding GP. As suggested by [4], GP performs no better than random search when expert knowledge is not provided. Thus, they cannot be scaled well with the increase in SNP size.

1.4 Paper Layout

This paper is organized as follows: the proposed methods are detailed in Section 2; data simulation and analysis are reported in Section 3; experimental results on simulated data are investigated in Section 4; experimental results on a real GWAS dataset of Rheumatoid Arthritis are illustrated in Section 5; the whole article is discussed and concluded in Section 6.

2 A Novel Discriminative Model: GP-Pi

In this research, we would like to design a new GP algorithm for discriminative modeling by utilizing expert knowledge. We have specifically developed an attribute weighting initializer which makes use of ReliefF [15] as expert knowledge. Although similar work [12] has been used in attribute selection approach (which used Tuned ReliefF), research has yet to prove if it works in discriminative modeling approaches. To enhance the interpretability of our model, we chose not to follow [12] to adopt MDR [16], an exhaustive feature construction algorithm, as one of the function sets. To prevent premature convergence to local optima, we maintain the diversity of population by penalizing similar individuals in their fitness. The introduction of a penalizing term in the fitness function is a novel GP application in GWAS. We evaluate our algorithm, GP-Pi (P stands for penalization and I stands for initialization), by comparing it against GPAS, which is publicly available at `http://ls2-www.cs.uni-dortmund.de/~nunkesser/`. Experimental results suggest that GP-Pi outperforms GPAS with statistically significance.

2.1 Genetic Programming Methods

We have adopted Genetic Programming to evolve models to model SNP-SNP interactions. We have used a crossover probability of 0.9, a mutation probability of 0.05, and a no-operation probability of 0.05. The maximum depth of a tree is 5. We have two function sets (And, Or) and two terminal sets (Equal, Not Equal). The overall GP parameters are summarized in Table 1. Comparing to the previous work, we contributed in two aspects. First, we introduced a penalizing term in the fitness function to penalize tree with common SNPs. Second, we developed an initializer which utilizes expert knowledge to initialize a good population. GP-Pi has been implemented in ECJ [17].

Tree Representation of Solutions. We have followed the approach suggested by [3] to express SNP-SNP interactions in logic expressions. However, we do not force the logic expression to be disjunctive normal form (DNF). Figure 1 depicts how we model GP expression trees to SNP-SNP interactions.

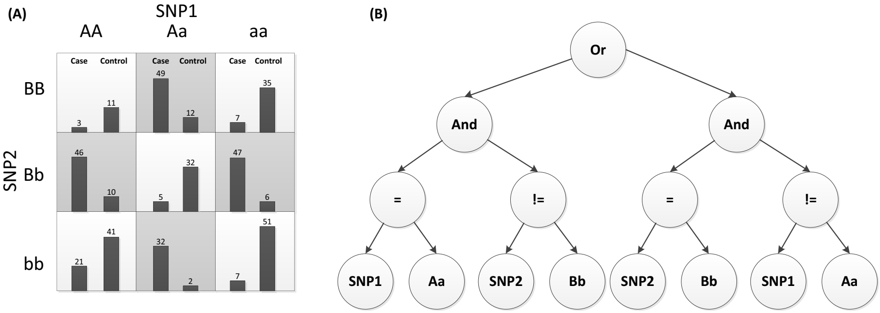

Fig. 1. (A) illustrates a SNP-SNP interaction in a GWAS. The left bars within each cell represent the number of cases (with diseases) while the right bars represent the number of controls (without diseases). Dark-shaded cells are high risk while the light-shaded cells are low risk. (B) is a GP tree modeling the predictive logic of the dark-shaded cells (high-risk combinations of SNP patterns). Hence, GP algorithm can be designed to evolve expression trees like (B) to model SNP-SNP interactions like (A).

Table 1. Summary of GP Parameters

Item	Parameter
Population Size	4096
Generations	40
Crossover	Single-point subtree
Crossover frequency	0.9
Mutation frequency	0.05
Fitness Function	$f_i = E_i + \frac{N_i}{\alpha} + P_i$
Selection	Sevens tournament
Function Set	And, Or
Terminal Set	$=, \neq$
Maximum Tree Depth	5

Initialization. We have developed a probabilistic initializer which utilizes expert knowledge to select attributes (SNPs). Here, we describe the mechanism of the initialization. First, we compute a score S for each attribute m in tournament. The attribute m with the highest score is selected. The equation is shown below:

$$S_m = R_m + U_m + \beta \times K \tag{1}$$

where R_m is the ReliefF score (See section 3), U_m is the usage score, β is a constant determined in runtime. If a random number V is larger than a threshold t_{relief} (0.8 is default), β will be 6, otherwise 1. K is a random number which is randomized in each tournament. Here, all random numbers range from 0 to 1 inclusively. The usage score is to keep the population as diverse as possible. The more an attribute appearing in the population, the lower its score is. The equation of Usage score is shown below:

$$U_m = \frac{L - u_m}{L} \tag{2}$$

where L is the highest number of appearance among all attributes and u_m is the number of appearance of m.

Penalization. Referencing a Koza style [18] of fitness function defined in [14], we introduced a penalization term to penalize trees with a certain degree of common SNPs and with similar number of nodes. The fitness function is shown below:

$$f_i = E_i + \frac{N_i}{\alpha} + P_i \tag{3}$$

where i is an index to an individual, E_i is the classification error, N_i is the number of nodes, α is the parsimony constant [19] (2 is default) and P_i is the penalization term. P_i is set to 100 if the following criterion is satisfied:

$$\frac{|N_i - N_j|}{m} < t_1 \wedge \frac{c}{m} > t_2 \tag{4}$$

where t_1 is a threshold (0.1 is default), t_2 is another threshold (0.9 is default), j is any individual within the population except i, c is the number of SNPs that are included by the tree expressions of both i and j, m is the average number of nodes between i and j. if the above criterion is not satisfied, P_i is set to 0.

In other words, every individual is compared against each other in every generation. Similar individuals are penalized based on the above procedure. This process helps maintain a wide diversity of population and prevents against premature convergence.

2.2 Expert Knowledge Guiding the Search

The use of expert knowledge is to provide an external measure of attribute quality to guide our search to overcome the needle-in-a-haystack problem [4]. Relief [15] is one feature selection algorithm which has the capability. The basic idea of Relief is to iteratively estimate feature weights according to their ability to discriminate between neighboring patterns. ReliefF [20] is an improvement in robustness of relief by considering the nearest k neighbors but not only the nearest one. Both Relief and ReliefF can capture attribute interactions because they use the entire vector of values to find nearest neighbor(s). However, they are also susceptible to noise attributes. Tuned ReliefF [21] is an improvement in susceptibility to noise attributes of ReliefF by systematically remove the low quality attributes so that the ReliefF score of the remaining attributes can be reestimated. Throughout this research, we adopted ReliefF but not Tuned ReliefF as our expert knowledge provider, for similar performance but less computational time. This is different from the similar work [12] which adopted Tuned ReliefF.

3 Data Simulation and Analysis

A simulation study was performed to evaluate our GP algorithm in Genome-Wide Association Study (GWAS). Using GAMETES [22], a GWAS dataset generator, we developed 4 penetrance functions (i.e. genetic model) which define a probabilistic relationship between genotype and phenotype (or the susceptibility to disease that depends only on genotype but not any other effects). Each model has 2 functional SNPs with minor allele frequency of 0.2. Also, each model is corresponded to a heritability (the effect size of functional SNPs) of 0.1, 0.2, 0.3, 0.4 respectively. An example is shown in Table 2. Based on the above model, each pair of functional SNPs was then combined within a genome-wide set of 98, 998 randomly generated SNPs to form a total of 100, 1000 attributes. Keeping a balanced ratio of cases and controls among 2000 samples, we generated 10 replicates for each parameter setting. A total of 80 datasets were generated and analyzed. All datesets with full precision are available upon request.

For each dataset, we ran our GP algorithm independently 10 times. For each parameter setting, we had in total 100 runs (10 replicates × 10 runs). For each parameter setting, we counted the number of times that the correct two functional SNPs were selected as nodes in the best GP tree model. This count was

Table 2. An Example Epistasis Model with Heritability 0.3

	AA (0.64)	Aa (0.32)	aa (0.04)
BB (0.64)	0.515	0.913	0.779
Bb (0.32)	0.934	0.124	0.383
bb (0.04)	0.614	0.712	0.792

expressed as a percentage, which was an estimation of the power of the method. Based on this count, we can estimate how often our GP algorithm can get the right answer if there is. We compared our result against GPAS (Genetic Programming for Association Study), a GP algorithm similar to ours but neither does it exploit expert knowledge nor penalize individuals. For each dataset, we ran GPAS on all simulated datasets with SNP size of 100 and 1000 for 10 independent runs, in total 100 runs for each parameter setting. 100,000 generations were allowed on each run. We consider the output of each run of GPAS as correct if the best 5 individuals contain the two functional SNPs. The power, i.e. the percentage that the algorithm outputs correct result, can then be estimated for GPAS. We compared the power of our GP algorithm against GPAS on estimation of power using chi-square test of independence. Results are considered statistically significant when the p-value of the chi-square test statistic is ≤ 0.05.

4 Experimental Results on Simulated Data

The power (the percentage that the algorithm identifies the correct two functional SNPs) for each method across heritability of 0.1, 0.2, 0.3 and 0.4 with a SNP size of 100 and 1000 is summarized in Figure 2. Each bar on the plots represents the power over the 100 runs of each parameter setting, or, in other words, represents the percentage that the algorithm can select the two functional SNPs as nodes of trees. Our algorithm was compared against GPAS in these experiments.

While our algorithm had a robust performance on 100 SNP size across different heritiabilities, it also had a satisfactory performance on 1000 SNP size. On 100 SNP size, we nearly achieved 100% power. On 1000 SNP size, we still had a certain percentage of power even when the heritability dropped to 0.1. In terms of comparison, our algorithm outperformed GPAS in terms of power on both SNP size of 100 ($P = 5.16E\text{-}10 < 0.05$) and SNP size of 1000 ($P = 4.85E\text{-}36 < 0.05$). The size of the search space is approximately 5000 (100 SNPs choose 2) and 500,000 (1000 SNPs choose 2). With a population size of 4096 and 40 generations, GP-Pi is exploring at most 3300% (SNP Size: 100) and 33% (SNP Size: 1000) of the search space. It should be noted that GPAS is allowed to have 100,000 generations on each run. Assume it has a population size of 10, GPAS is exploring at most 20000% (SNP: 100) and 200% (SNP: 1000) of the search space. While the best 5 individuals on each run of GPAS are extracted to examine its correctness, only the best individual is outputted to be examined in GP-Pi.

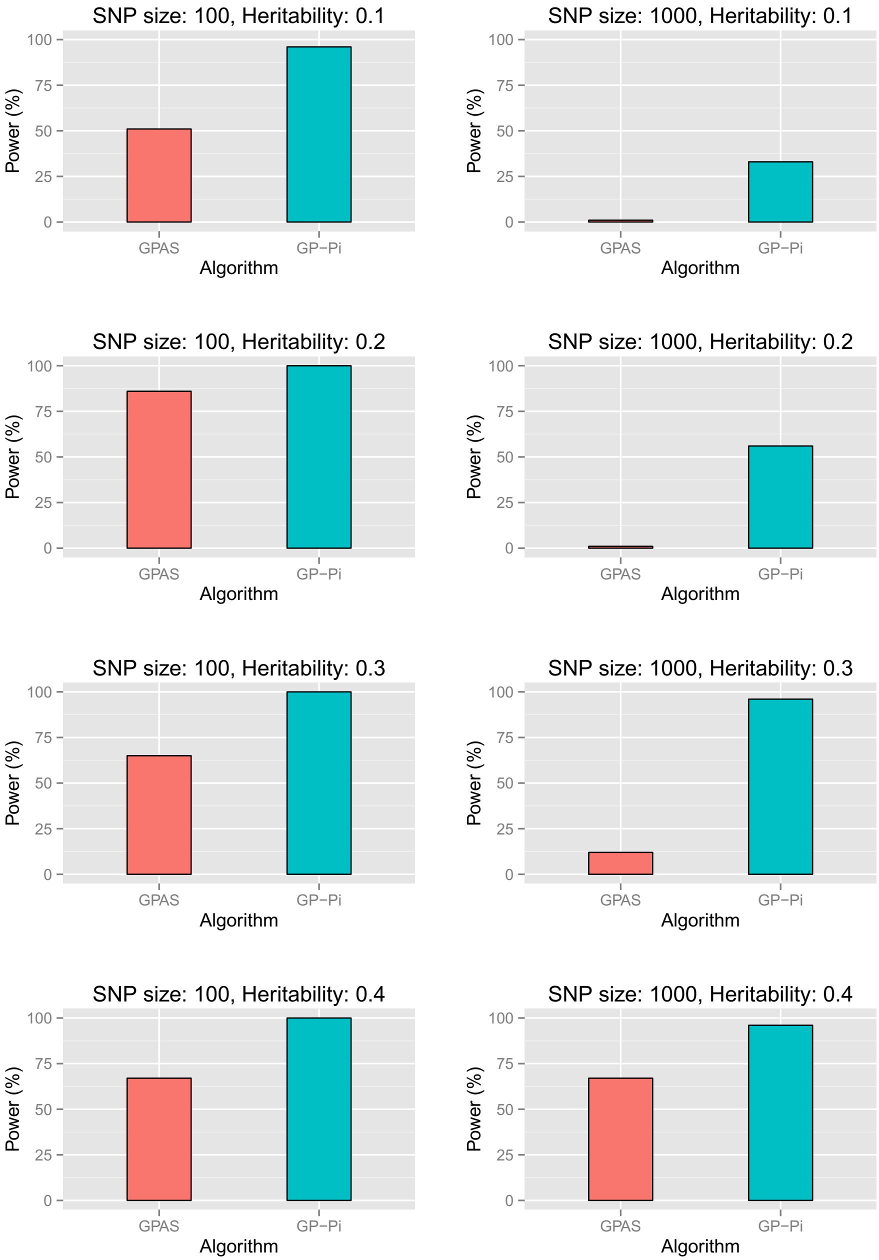

Fig. 2. The power (the percentage that the algorithm identifies the correct two functional SNPs) of GPAS and GP-Pi across heritability of 0.1, 0.2, 0.3 and 0.4 with a SNP size of 100 and 1000 is summarized above. GP-Pi outperformed GPAS in terms of power on both SNP size of 100 (P = 5.16E-10 <0.05) and SNP size of 1000 (P = 4.85E-36 <0.05).

5 Experimental Results on Real Data

In this section, we demonstrate the capability of GP-Pi on handling with real data. We applied GP-Pi on a real GWAS dataset. Our GWAS dataset is obtained from the North American Rheumatoid Arthritis Consortium (`http://www.naracdata.org/`). There are 2062 patients in total, 868 of which are cases which suffer from Rheumatoid Arthritisand the remaining (1194) are controls. There are in total 545080 SNPs in this study, which constitutes a gigantic search space.

5.1 Data Preprocessing

There are three types of genotypes: 'AA', 'Aa' and 'aa'. They were encoded as 1, 2, 3 respectively. SNPs with more than 15% of missing values were filtered. The missing values of the remaining SNPs were chosen randomly. As the number of SNPs in our search space is tremendous, we select the top 1000 SNPs with top ReliefF scores.

5.2 Data Mining and Parameter Setting

The dataset was shuffled and split into testing data and training data on a ratio of 1 to 9. GP-Pi was run on training data under a 10-fold cross-validation to learn classification models. In total, 10 classification models were learned and evaluated on testing data. It should be noted that the same set of GP parameters, summarized in Table 1 were used in these experiments.

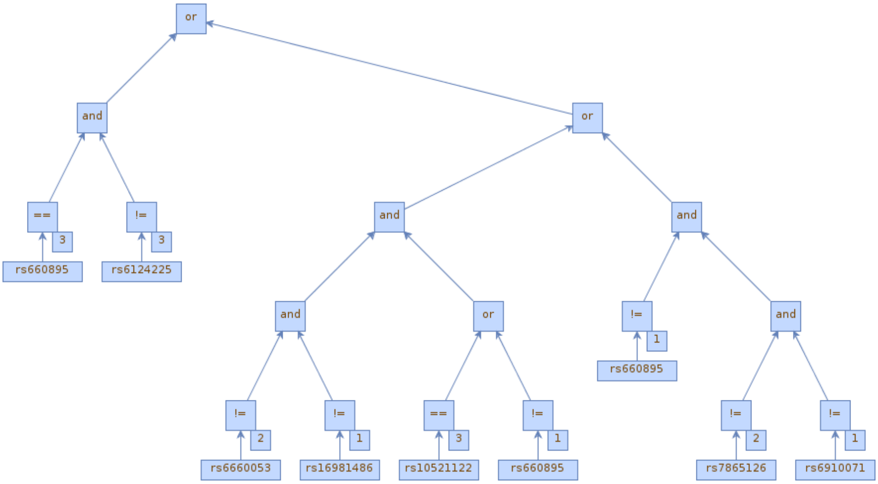

Fig. 3. The above is the best discriminative model outputted by GP-Pi on a real GWAS dataset of Rheumatoid Arthritis to classify if a patient is a case or a control. It has achieved a sensitivity of 0.741 and a specificity of 0.806. Both rs660895 and rs6910071 are confirmed to be risk loci with literature support respectively [23, 24]. rs7865126, with its susceptibility to tuberculosis infection [25, 26], may be a new discovery.

5.3 Result and Analysis

Among the 10 models, the best one is selected based on its performance in validation data . It is then evaluated on testing data and the result is as follows: Accuracy: 0.790, Precision: 0.741, Recall: 0.769, F-measure: 0.754, Specificity: 0.806. The best model is demonstrated in Figure 3. It should be noted that both rs660895 and rs6910071 are confirmed to be rheumatoid arthritis risk loci with literature support respectively [23, 24]. In addition, rs7865126 may be a novel discovery of rheumatoid arthritis risk loci.

6 Discussion and Conclusion

In this paper, our objective is to show discriminative modeling on GWAS with GP can be improved by utilizing expert knowledge. We have developed an initializer which exploits ReliefF score as expert knowledge to seed the initial population with good attributes, and introduced a penalization term into the fitness function to penalize trees with too many common SNPs. Experiments have been run on simulated data and compared against our performance with GPAS. Results on simulated data has shown that GP-Pi outperformed GPAS with statistical significance, where GPAS is a discriminative modeling GP with neither penalization nor initialization. It shows that our method plays an important role in improving performance on guiding the search.

We also have applied GP-Pi on a real GWAS dataset of Rheumatoid Arthritis to prove the applicability of GP-Pi. Our method performed up to expectation in that (1) it picked up both HLA SNPs which are known to have the largest effect on RA susceptibility (rs660895 and rs6910071), (2) it demonstrated the ability of the method to delineate both local and inter-chromosomal interaction effects. For local interaction, it represents the effect of haplotype and is exemplified by the rightmost branch (involving rs660895, and rs6910071). In addition, this branch also involves an interaction with rs7865126 which is located in another chromosome. The leftmost branch is another example of inter-chromosomal (inter-genic) interaction which involves rs660895 (a HLA gene) and rs6124225. The decision tree also showed internal consistence, for example, in all branches where rs660895 was involved, either ”3” or !=”1” genotypes were found to associated with disease. In addition, rs7865126 was found to be a new interaction partner with SNPs in HLA loci. rs7865126 is located within the gene coding for a subunit of the augmin complex. Recently, Png et al [25] and Li et al [26] reported SNPs in this gene are related to susceptibility to tuberculosis infection and possibly to innate response.

In this paper, we have shown that discriminative modeling on GWAS with GP can be significantly improved by penalization and initialization . We have also illustrated that GP performs not only attribute selection but also discriminative modeling well. Our next question is about the limits of these algorithms. Can these algorithms process millions of genetic variations directly without any kind of filtering? If not, what are the alternatives? These questions may be challenging to answer. However, we believe that with the exploding data volume of human genetics, intelligent algorithms like GP will be in critical need and keep on playing an important role in the future development of GWAS.

References

1. Hirschhorn, J., Daly, M.: Genome-wide association studies for common diseases and complex traits. Nature Reviews Genetics 6(2), 95–108 (2005)
2. Wang, W., Barratt, B., Clayton, D., Todd, J.: Genome-wide association studies: theoretical and practical concerns. Nature Reviews Genetics 6(2), 109–118 (2005)
3. Nunkesser, R., Bernholt, T., Schwender, H., Ickstadt, K., Wegener, I.: Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. Bioinformatics 23(24), 3280–3288 (2007)
4. Moore, J., White, B.: Genome-wide genetic analysis using genetic programming: The critical need for expert knowledge. In: Genetic Programming Theory and Practice IV, pp. 11–28 (2007)
5. Reich, D., Lander, E.: On the allelic spectrum of human disease. TRENDS in Genetics 17(9), 502–510 (2001)
6. Moore, J., Asselbergs, F., Williams, S.: Bioinformatics challenges for genome-wide association studies. Bioinformatics 26(4), 445–455 (2010)
7. Martin, M.C.: Genetic programming for real world robot vision. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 67–72. IEEE (2002)
8. Chen, S.H.: Genetic algorithms and genetic programming in computational finance. Springer (2002)
9. Langdon, W., Barrett, S.: Genetic programming in data mining for drug discovery. In: Evolutionary Computation in Data Mining, pp. 211–235 (2005)
10. Lo, L., Chan, T., Lee, K., Leung, K.: Challenges rising from learning motif evaluation functions using genetic programming. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 171–178. ACM (2010)
11. Wong, K., Peng, C., Wong, M., Leung, K.: Generalizing and learning protein-dna binding sequence representations by an evolutionary algorithm. Soft Computing-A Fusion of Foundations, Methodologies and Applications 15(8), 1631–1642 (2011)
12. Greene, C., White, B., Moore, J.: Sensible initialization using expert knowledge for genome-wide analysis of epistasis using genetic programming. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1289–1296 (2009)
13. Greene, C., White, B., Moore, J.H.: An expert knowledge-guided mutation operator for genome-wide genetic analysis using genetic programming. In: Rajapakse, J.C., Schmidt, B., Volkert, L.G. (eds.) PRIB 2007. LNCS (LNBI), vol. 4774, pp. 30–40. Springer, Heidelberg (2007)
14. Estrada-Gil, J., Fernández-López, J., Hernández-Lemus, E., Silva-Zolezzi, I., Hidalgo-Miranda, A., Jiménez-Sánchez, G., Vallejo-Clemente, E.: Gpdti: A genetic programming decision tree induction method to find epistatic effects in common complex diseases. Bioinformatics 23(13), i167–i174 (2007)
15. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proceedings of the Ninth International Workshop on Machine Learning, pp. 249–256. Morgan Kaufmann Publishers Inc. (1992)
16. Hahn, L., Ritchie, M., Moore, J.: Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. Bioinformatics 19(3), 376–382 (2003)

17. Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Bassett, J., Hubley, R., Chircop, A.: Ecj: A java-based evolutionary computation research system (2007)
18. Koza, J., James, P.: Rice, genetic programming (videotape): the movie (1992)
19. Bleuler, S., Brack, M., Thiele, L., Zitzler, E.: Multiobjective genetic programming: Reducing bloat using spea2. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 536–543. IEEE (2001)
20. Wiskott, L., Fellous, J., Kruger, N., Malsburg, C.: Estimating attributes: analysis and extension of relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
21. Moore, J.H., White, B.C.: Tuning relieff for genome-wide genetic analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)
22. Urbanowicz, R., Kiralis, J., Sinnott-Armstrong, N., Heberling, T., Fisher, J., Moore, J.: Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. BioData mining 5(1), 16 (2012)
23. Gorman, J., David-Vaudey, E., Pai, M., Lum, R., Criswell, L.: Particular hla–drb1 shared epitope genotypes are strongly associated with rheumatoid vasculitis. Arthritis & Rheumatism 50(11), 3476–3484 (2004)
24. Stahl, E.A., et al.: Genome-wide association study meta-analysis identifies seven new rheumatoid arthritis risk loci. Nat Genet. 42(6), 508–514 (2010)
25. Png, E., Alisjahbana, B., Sahiratmadja, E., Marzuki, S., Nelwan, R., Balabanova, Y., Nikolayevskyy, V., Drobniewski, F., Nejentsev, S., Adnan, I., et al.: A genome wide association study of pulmonary tuberculosis susceptibility in indonesians. BMC Medical Genetics 13(1), 5 (2012)
26. Li, S., Wang, L., Berman, M., Kong, Y.Y., Dorf, M.E.: Mapping a dynamic innate immunity protein interaction network regulating type i interferon production. Immunity 35(3), 426–440 (2011)