

EPIRL: A REINFORCEMENT LEARNING AGENT TO FACILITATE EPISTASIS DETECTION

Kexin Huang

Courant Institute
New York University
521 Mercer Street, New York, NY 10012
kexin.huang@nyu.edu

Rodrigo Nogueira

Tandon School of Engineering
New York University
6 MetroTech Center, Brooklyn, NY 11201
rodrigonogueira@nyu.edu

ABSTRACT

Epistasis (gene-gene interaction) is crucial to predicting genetic disease. Our work tackles the computational challenges faced by previous works in epistasis detection by modeling it as a one-step Markov Decision Process where the state is genome data, the actions are the interacted genes, and the reward is an interaction measurement for the selected actions. A reinforcement learning agent using policy gradient method then learns to discover a set of highly interacted genes.

1 INTRODUCTION

The fundamental goal for studying genetics is to understand how certain genes can incur disease and traits. Since the advent of Genome-Wide Association Studies (GWAS) (Burton et al., 2007), thousands of SNP (Single Nucleotide Polymorphism)s have been identified and associated with genetic diseases and traits. These SNPs are discovered through one-SNP-at-a-time statistical analysis. However, individual gene marker is insufficient to explain many complex diseases and traits (Mackay & Moore, 2014). Now, most geneticists believe that gene-gene interaction (epistasis) can explain the missing heritability incurred by the traditional approach.

There has been a substantial amount of work on epistasis detection. Exhaustive combinatorial search methods like Multifactor Dimensionality Reduction (MDR) (Yang et al., 2018) have been shown successful, but only in small genome-scale due to computational complexity. Later, attempts to reduce search spaces exhibit efficiency, like ReliefF and Spatially Uniform ReliefF (Niel et al., 2015). Besides, machine learning-based algorithms gain popularity. Most of the machine learning based methods for epistasis detection model the epistatic process as a non-linear neural network. It predicts if an input sequence is disease or healthy. Then, they rely on examining the internal weights of the models to find the interacting SNPs. A high weight means the corresponding SNPs are contributing to predict the disease. If multiple high weight SNPs are detected, then they are considered interacted. For example, Random Forest models each node as an SNP and grows a classification tree and later examines the decision trace for interpretation (Jiang et al., 2009); BEAM (Bayesian Epistasis Association Mapping) uses MCMC algorithm to iteratively test each marker’s probability of association with disease, dependent on other markers (Zhang et al., 2011). However, most machine learning based algorithms suffer from a limited number of input sequences compared to the size of the sequence (#SNPs). Another interesting approach is ant colony optimization algorithm (Wang et al., 2010), which finds a refined subset of SNPs by iteratively updating a selection probability distribution.

Although there are efficient methods to measure if a given SNPs set interact, previous works all suffer from the high computational cost of finding all possible n -combinations of SNP. For example, for a standard GWAS dataset with 10^6 SNPs, a 2-locus exam requires $5 * 10^{11}$ searches, a 3-locus exam asks for $1.6 * 10^{17}$, a 4-locus search needs $4 * 10^{22}$ iterations. Hence, how to utilize these metrics to get an SNP set from a genome-scale data is the challenging part. Another challenge is that all the algorithms above assume and output fixed n -locus interactions (typically 2 or 3) where n is unknown for real biological data. We tackle these two challenges by introducing a novel model based on Reinforcement Learning to the task of epistasis detection.

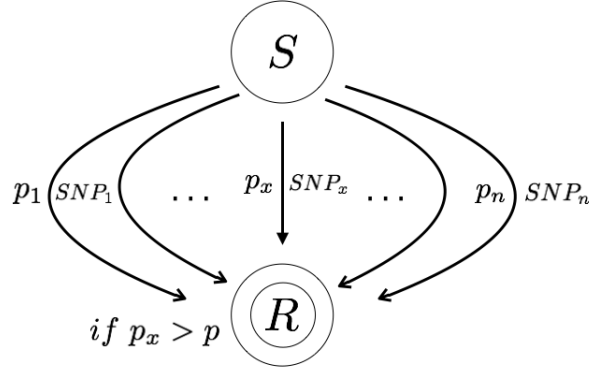


Figure 1: An illustration of our One-Step MDP Model. S is genome data, and the state has l actions where a probability p_x is associated with the action SNP_x . All SNP_x whose p_x are larger than a threshold p would be selected as an interaction set and reward R is computed based on the set. Then the One-Step MDP terminates.

2 METHOD

2.1 MODEL

A typical GWAS dataset contains examples of sequences with no disease (control) and with disease (case), where both have l SNPs. We denote t_1 and t_2 as the number of control and case sequences, respectively. Each SNP has three genotypes $\{aa, Aa, AA\}$, which is encoded by $\{0, 1, 2\}$. We want to find a set of highly interacted SNPs with the size from 2 to n .

We model the epistasis process as a one-step Markov Decision Process (MDP) (Figure 1). The state S is a latent representation encoded from genome data; The action space is all the SNPs, where highly interacted SNPs are selected by a probability threshold so that it poses no constraint to fix the size of interaction; the reward is efficient interaction measurements like MDR correct classification rate (CCR) and Rule Utility (Yang et al., 2018). A reinforcement learning agent will learn to select SNPs that have high rewards, i.e., high interaction, by using the policy gradient method. Our approach solves the challenges mentioned above because firstly since it optimizes over iterations and chooses only a small set of actions, it is non-exhaustive, which means computationally feasible. It also utilizes the efficacy of interaction measurements like MDR CCR and Rule Utility. Second, it picks an action as long as the action passes a probability threshold, which means it can output a different size of interaction set every iteration.

2.2 NETWORK

For the input D , we randomly sample K sequences, half from the case and the other half from the control data set (Figure 2). We then encode each sequence using the output of Convolutional Neural Network (CNN) or last hidden state of Recurrent Neural Network (RNN) to capture the spatial structure of the genome. These K latent representations will be the state for our EpiRL agent.

We then feed the state into a two-layer neural network W , which serves as a value function approximator. The neural network will output l probabilities $P(SNP_m|D)$ for every SNP. We determine the size of interactions n as the number of SNPs that have probabilities larger than $1/n$ to allow up to n -locus interaction. We then sample n SNPs based on the probability distribution generated by the network to ensure exploration for our RL agent. This filtering forms our interaction set $I = \{SNP_{a_1}, SNP_{a_2}, \dots, SNP_{a_n}\}$.

2.3 REWARD

Given this SNP set I , we calculate the reward, which measures the interaction. Our method uses the sum of two metrics as a reward: MDR CCR and Rule Utility (Yang et al., 2018). These two measures are based on MDR (Motsinger & Ritchie, 2006), which is a procedure that collapses the selected

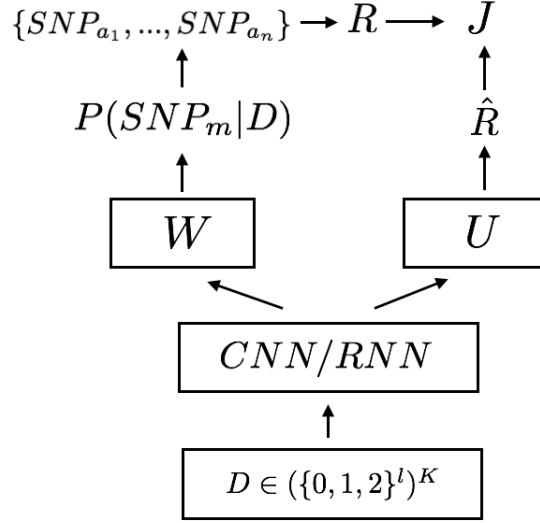


Figure 2: An illustration of our EpiRL agent. The agent first encodes D , a mini-batch of genome data, and then it predicts action-values through W where a set of actions are selected and reward calculated. Along with the baseline reward, we compute the loss and iterate to learn the best actions with Reinforcement Learning.

interacted data set into a four variable table. Then we perform two statistical calculations on top of this table, described as follows.

We have a genome data with size $(t_1 + t_2) \times l$, where t_1 and t_2 are the number of sequences in control and case, respectively, and l is the total number of SNPs. Suppose our RL agent picks n actions. We then extract the genome data with these selected SNPs and form a sub data set I with size $(t_1 + t_2) \times n$.

There are three genotypes for each SNP: $\{0, 1, 2\}$. Therefore, for n SNPs, there are 3^n possible combinations of SNPs. We denote each combination as α_t where $t \in \{1, \dots, 3^n\}$. For each combination, we can count the number of control and case in I . We denote them α_t^1 and α_t^2 . We assign a binary category to each combination: if $\frac{\alpha_t^2}{\alpha_t^1} < 1$, then this combination is in the low-risk group LR , and if ≥ 1 , it is called a high-risk group HR . Basically, for this specific genotype, if the number of the case exceeds the number of control, then it is high-risk, and vice versus. Now, we can construct four variables:

	High Risk	Low Risk
Case	TP	FN
Control	FP	TN

where

$$TP = \sum_{t \in HR} \alpha_t^2 \quad (1)$$

$$FP = \sum_{t \in HR} \alpha_t^1 \quad (2)$$

$$FN = \sum_{t \in LR} \alpha_t^2 \quad (3)$$

$$TP = \sum_{t \in LR} \alpha_t^1 \quad (4)$$

The above equations mean that we first divide case and control sequences in the low and high-risk group and then retrieve the number of cases and controls in each group. Now, we can calculate the two

measures. These two measures together are shown to be effective in measuring epistasis (Yang et al., 2018). MDR CCR is the correct classification rate and Rule Utility U derives from the chi-square statistics of rule relevance, which measures the interaction:

$$CCR = 0.5 \cdot \left(\frac{TP}{TP + FN} + \frac{TN}{FP + TN} \right) \quad (5)$$

$$U = \frac{(R - \delta)^2}{(1 + \delta)(\gamma - \delta - 1)} \quad (6)$$

where

$$R = \frac{FP + TN}{TP + FN} \quad (7)$$

$$\delta = \frac{FP}{TP} \quad (8)$$

$$\gamma = \frac{TP + FP + TN + FN}{TP} \quad (9)$$

We sum CCR and U as our reward. Note that the calculation is fast since n is usually a small number. In our preliminary study on a set with 100 SNPs, the average running time for one iteration is 0.01s, where an iteration consists that the network predicts probabilities, calculates the reward and back-propagates the gradient.

2.4 TRAINING

We train the model using REINFORCE algorithm (Williams, 1992). Our objective consists of three parts:

$$J_1 = (R - \hat{R}) \sum_{t \in I} -\log P(t|D) \quad (10)$$

$$J_2 = ||R - \hat{R}||^2 \quad (11)$$

$$J_3 = \lambda \sum_{t \in L} p(t|D) \log P(t|D) \quad (12)$$

\hat{R} is a baseline reward computed by the value network U , a 2-layer neural network that minimizes J_2 . J_1 is the advantage policy gradient. The advantage is the gap between reward and baseline, which ensures the agent to prefer actions that output rewards higher than expected. J_3 is the entropy regularization across all SNPs L to mitigate peaky probability distribution, where λ is the parameter to adjust the intensity of the mitigation.

3 EXPERIMENT

We use simulated data from GAMETES software, which generates random, pure, strict, n-locus epistasis model (Urbanowicz et al., 2012). To evaluate our method, we record SNPs set with top K rewards across C generated datasets. We compare this set with the ground truth labels and compute the recall $R@K = \frac{L}{C}$ where the agent gets L predictions right in C data set. We are also interested in the average time the agent takes to detect the right interaction.

In our preliminary study, we experiment our agent in a simulated 2-locus dataset with 600 sequences of the case and control set and with 100 SNPs. We design our data with standard genome constraint: 0.2 heritability; 0.7 prevalence; 0.2 minor allele frequency for both of 2 interacting SNPs. We minimize our objectives using the Adam optimizer (Kingma & Ba, 2014) with learning rate $1e^{-3}$.

We experimented the RL agent 50 times on the same data set. In each round of experiment, the RL agent is asked to find the interacted 2-locus SNPs under 5000 iterations. Out of the 50 trials, 34 times the agent finds the interacted SNPs under 5000 iterations. In the 34 times that the agent successfully predicts the interaction set, the average iteration is 2260.6 and the average time to find the SNPs is 22.4 s. In comparison, the exhaustive search takes 51 s.

In the future, we will experiment on a larger dataset with various locus interactions. We will compare the recalls and the average running time with existing methods: MDR, BEAM, and Ant Optimization. At last, we will run the agent on GWAS Coronary Artery Disease (CAD) dataset since CAD is shown under epistasis effect and we will compare other study's reported epistasis on CAD with ours.

4 CONCLUSION

Our work proposes a novel approach to model epistasis detection as a one-step MDP and introduces reinforcement learning to address this problem. We believe this will lead a new path to tackle the computational challenge in gene-gene interaction detection.

REFERENCES

- Paul R Burton, David G Clayton, Lon R Cardon, et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. 2007.
- Rui Jiang, Wanwan Tang, Xuebing Wu, and Wenhui Fu. A random forest approach to the detection of epistatic interactions in case-control studies. *BMC bioinformatics*, 10(1):S65, 2009.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Trudy FC Mackay and Jason H Moore. Why epistasis is important for tackling complex human disease genetics. *Genome medicine*, 6(6):42, 2014.
- Alison A Motsinger and Marylyn D Ritchie. Multifactor dimensionality reduction: an analysis strategy for modelling and detecting gene-gene interactions in human genetics and pharmacogenomics studies. *Human genomics*, 2(5):318, 2006.
- Clément Niel, Christine Sinoquet, Christian Dina, and Ghislain Rocheleau. A survey about methods dedicated to epistasis detection. *Frontiers in genetics*, 6:285, 2015.
- Ryan J Urbanowicz, Jeff Kiralis, Nicholas A Sinnott-Armstrong, Tamra Heberling, Jonathan M Fisher, and Jason H Moore. Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData mining*, 5(1):16, 2012.
- Yupeng Wang, Xinyu Liu, Kelly Robbins, and Romdhane Rekaya. Antepiseeker: detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm. *BMC Research Notes*, 3(1):117, Apr 2010. ISSN 1756-0500. doi: 10.1186/1756-0500-3-117. URL <https://doi.org/10.1186/1756-0500-3-117>.
- Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL <https://doi.org/10.1007/BF00992696>.
- Cheng-Hong Yang, Yu-Da Lin, and Li-Yeh Chuang. Multiple-criteria decision analysis-based multifactor dimensionality reduction for detecting gene-gene interactions. *IEEE Journal of Biomedical and Health Informatics*, 2018.
- Yu Zhang, Bo Jiang, Jun Zhu, and Jun S Liu. Bayesian models for detecting epistatic interactions from genetic data. *Annals of human genetics*, 75(1):183–193, 2011.