國立臺灣大學電資學院生醫電子與資訊學研究所
博士論文

Graduate Institute of Biomedical Electronics and Bioinformatics
College of Electrical Engineering and Computer Science
National Taiwan University
Doctoral Dissertation

開發以機器學習搭配基因資訊為基礎之上位作用偵測演算法
Gene-based Epistasis Discovery Using Machine Learning

張育銓
Yu-Chuan Chang

指導教授：歐陽彥正 博士、陳倩瑜 博士
Advisor: Yen-Jen Oyang, Ph. D.
Chien-Yu Chen, Ph. D.

中華民國 一〇九 年 一 月
January, 2020

doi:10.6342/NTU202000613

國立臺灣大學（碩）博士學位論文
口試委員會審定書

開發以機器學習搭配基因資訊為基礎之
上位作用偵測演算法
Gene-based Epistasis Discovery Using Machine Learning

本論文係張育銓君（學號 D00945009）在國立臺灣大學生醫電子與資訊學研究所完成之博士學位論文，於民國一〇九年一月二十日承下列考試委員審查通過及口試及格，特此證明

口試委員：

[illegible]（指導教授）

陳倩瑜（指導教授）

蔡懷寬

吳君泰

賴飛羆

系主任、所長 張瑞峰（簽名）

中文摘要

全基因體關聯性分析(Genome-wide association study)是一個偵測基因變異(Genetic variant)與外顯型(Phenotype)之間關聯性的常用方法。然而，全基因體關聯性分析在偵測基因變異之間的交互作用與外顯型的關聯性，或稱為上位作用(Epistasis)，的能力有限。我們認為，開發一個有效且有效率的全基因體關聯性分析方法來偵測上位作用，將有助於解開 像是阿茲罕默症(Alzheimer’s disease)等複雜疾病(Complex disease)的致病機制。因此，本研究開發一個演算法：GenEpi，此演算法利用機器學習(Machine learning)來偵測變異之間的交互作用與外顯型的關聯性。由於在生物學上，基因是最小的功能單位，故 GenEpi 的核心概念便是利用基因(Gene)在基因體中的區段為分割區塊，並分兩個階段進行特徵值的萃取，試圖解決偵測全基因體上位作用計算複雜度(Computational complexity)過高的問題，以及多重檢定(Multiple testing)導致統計信度下降的問題。GenEpi 的兩個階段分別為基因內(Within-gene)的上位作用偵測，以及基因間(Cross-gene)的上位作用偵測。在這兩個階段我們皆使用二元組合編碼(Two-element combinatorial encoding)來產生代表上位作用的特徵值，並利用正規化回歸(L1-regularized regression)以及穩定性選擇法(Stability selection)來篩選特徵值並建立模型。本研究將 GenEpi 運用於阿茲罕默症的資料集上，來預測樣本是否為阿茲罕默症病患或預測阿茲罕默症的病程快慢，藉此驗證演算法成效，並期待因此能進一步揭開更多阿茲罕默症可能的致病因子。不論是在模擬資料或是阿茲罕默症真實資料，結果顯示 GenEpi 的預測準確度及計算時間皆優於現行的演算法，如：FastEpistasis，BOOST，ReliefF 等方法。足見 GenEpi 將有助於其他全基因體關聯性分析，特別是針對複雜疾病的研究，預期將可提供生醫研究人員進行實驗設計時更多有用的參考資訊。

可用性：GenEpi 是一個開源的 Python 套件，授權給非商業行為的學術人員使用。原始碼已公開在 PyPi 套件庫，以及 GitHub (https://github.com/Chester75321/GenEpi)。

關鍵詞：GenEpi、機器學習、上位作用、全基因體關聯性分析、阿茲罕默症。

doi:10.6342/NTU202000613

Abstract

Genome-wide association studies (GWAS) provide a powerful means to identify associations between genetic variants and phenotypes. However, GWAS techniques for detecting epistasis, the interactions between genetic variants associated with phenotypes, are still limited. We believe that developing an efficient and effective GWAS method to detect epistasis will be a key for discovering sophisticated pathogenesis, which is especially important for complex diseases such as Alzheimer’s disease (AD). In this regard, this study presents GenEpi, a computational package to uncover epistasis associated with phenotypes by the proposed machine learning approach, and illustrates the application of GenEpi on predicting the diagnosis and the progression of AD. The key concept of GenEpi is a two-stage feature extraction process based on gene structures. Since a gene is the minimal physical and functional unit of heredity, GenEpi considers a gene as a unit to retrieve genetic variants as features. GenEpi adopts two-element combinatorial encoding when producing features and constructs the prediction models by L1-regularized regression with stability selection. Features are first modeled using combinatorial encoding followed by L1-regularized regression with stability selection to detect the epistasis within a single gene. The selected features for each gene are then pooled together to identify cross-gene epistasis, using L1-regularized regression with stability selection again. This study compared GenEpi with several commonly used algorithms for detecting epistasis, including FastEpistasis, BOOST and ReliefF. The simulation data demonstrated that GenEpi outperforms the other methods in ranking the true epistasis as the top one. As real data is concerned, the results suggested that the epistasis selected by GenEpi has the best predictive power for two major phenotypes in the AD dataset: diagnosis of AD and disease progression. For diagnosis, the proposed model of predicting AD contains three clinical features (Age, Gender and Education) and

doi:10.6342/NTU202000613

14 genetic features, including 24 SNPs from 12 genes that contain the well-known causal gene, *APOE*. The 2-fold cross validation (CV) and leave-one-out CV (LOO CV) accuracy of this model are 0.83 and 0.81, respectively. On the other hand, for predicting progression, the proposed model contains eight clinical features (Age, Gender, Education, Cognitively Normal (CN), Early Mild Cognitive Impairment (MCI), Late MCI, AD, and MMSE at baseline) and four genetic features, including seven SNPs from six genes, where all of the four genetic features are cross-gene epistasis with significant p-values ($< 10^{-11}$). The average of Pearson and Spearman correlation of 2-fold CV and LOO CV are 0.52 and 0.53, respectively. The results on AD revealed the capability of GenEpi in finding disease-related variants and epistasis that show both biological meanings and predictive power. The released package can be generalized to largely facilitate the studies of many complex diseases in the near future.

Availability: GenEpi is an open-source Python package and available free of charge for non-commercial users. The package has been published on The Python Package Index, and GitHub (https://github.com/Chester75321/GenEpi)

Keywords: GenEpi, Machine Learning, Epistasis, GWAS, Alzheimer’s disease

Table of Contents

doi:10.6342/NTU202000613

List of Figures

doi:10.6342/NTU202000613

List of Tables

doi:10.6342/NTU202000613

1. Introduction

Genome-wide association studies (GWAS) is a univariate examination of a genome-wide set of genetic variants to determine if any single variant is associated with the phenotype of interest [1]. The first GWAS was published in 2002 [2], and three years later, the most remarkable GWAS regarding age-related macular degeneration (AMD) was published [3]. Their study investigated the association of 105,980 single nucleotide polymorphisms (SNPs) with AMD on 96 cases and 50 control subjects. This study showed that the SNPs in the complement factor H (*CFH*) gene, including a non-synonymous SNP, are significantly associated with AMD. Up to 2019, there have been more than hundreds of thousands individuals being studied in typical GWAS protocols, and over 210,498 variant-disease associations between 117,337 SNPs and 10,358 phenotypes have been discovered [4]. These studies demonstrated the potential of GWAS to identify genetic variants associated with many categories of phenotypes, including risks for diseases such as various cancers, and variations in therapeutic and adverse responses to drugs. However, the success of univariate GWAS is limited to monogenic phenotypes (e.g. Mendelian diseases). The impact of variant interactions, also known as epistasis on the formation of diseases [5] is often underestimated in traditional GWAS analysis [6-8].

A major limitation of traditional GWAS is that it considers only one genetic variant at a time, and ignores underlying epistasis of variants that might have stronger associations [9]. Researchers have found that GWAS has limitation in identifying the association in complex diseases [10, 11]. Easton *et al.* suggested that a number of susceptible loci identified by GWAS usually have very small effect sizes [12]. Studies have also demonstrated that the existence of epistasis is an important factor contributing to phenotypes, especially in complex diseases such as hypertension, diabetes and obesity [11]. Therefore, developing analytical methods to identify epistasis efficiently is critical

doi:10.6342/NTU202000613

to understanding the genetic factors [8, 13], and has attracted a wide range of research interests in recent years [7, 14].

There are, however, two main challenges to discover epistasis: computational complexity and statistical power [15]. The first challenge results from the curse of dimensionality. When more genetic variants are considered, the number of interactions increases exponentially. Based on the specification of a major commercial technology, Illumina Arrays, a whole-genome array can investigate over 4 million markers per sample simultaneously. In order to evaluate the pairwise interactions from this microarray, about 8×10^{12} statistical tests need to be processed. Even though Marchini *et al.* have demonstrated that pairwise interactions of 3×10^{5} loci is computationally possible with currently available computational resources, it still remains challenging when the Illumina Arrays are considered [16]. The second challenge is the issue of statistical power. Since a huge number of statistical tests are conducted on a limited sample size with high-dimensional interactions, many false positives arise by random chances. In recent years, new methods have been developed to tackle the issue of epistasis [11, 17]. Statistical approaches include FastEpistasis [18] and BOOST [19]; both of them has been included in a well-known GWAS software called PLINK [20, 21]. Machine learning approaches such as Multifactor Dimensionality Reduction [22], ReliefF [23], random forest-like algorithms [24-26] and other methodologies have also been developed for detecting epistasis [17].

Since the biological experiments used to validate these methodologies are still in demand, there are no standard analysis methods for epistasis despite the rapid improvement in computational performance. In 2016, Murk used FastEpistasis and BOOST to search SNP-SNP interactions on a huge dataset called Genetic Epidemiology Research on Adult Health and Aging (GERA) that included 78,486 subjects, but still

doi:10.6342/NTU202000613

failed to detect a significant and replicable interaction after exhaustively searching through 45 billion possible interactions for 10 complex diseases of interest [27]. Alzheimer's disease (AD) is one of the most important complex diseases and its pathogenesis, which clearly has a genetic basis, is still ill-defined. In 2014, Sage Bionetworks held a competition called The Dialogue for Reverse Engineering Assessments and Methods Challenge (DREAM Challenge) for AD, which tried to use crowdsourcing to assess the capability of current computational methods to predict the change in cognitive examination based on genetic data. However, no significant contribution of genetic features except the *APOE* haplotype to the predictive performance was observed by any competition teams [28]. In order to discover more SNP interactions with both statistical and biological significance, this study presents GenEpi, a package to reveal epistasis related to the phenotype using machine learning and introduces the application of GenEpi on AD.

doi:10.6342/NTU202000613

2. Related Works

2.1. Introduction of Alzheimer's Disease

There are about 66 million people around the world who live with dementia and the number might increase to 115 million by 2050. Alzheimer's disease (AD) is one of the leading causes for dementia in the elderly. Clinically, AD can be divided into early-onset (patients younger than 65 years) and late-onset (patients older than 65 years). The presence of plaques of amyloid β peptides and intra-neuronal tangles of hyper-phosphorylated forms of microtubule-associated protein tau (*MAPT*) can be found in the patient's brain [29]. Instead of these two phenomena, genetic factors are also important for AD. The mutations in three genes, *Presenilin 1*, *Presenilin 2* and amyloid precursor protein (*APP*), have been found to be the causes of the early-onset forms of AD [30-33]. It is generally believed that the late-onset form of AD is most likely caused by the combination of several genetic and environmental factors, the so-called complex diseases [29]. The most famous genetic risk factor, the *APOE* ε4 allele, has already been implicated for AD. As technology advances, such as using GWAS, genetic risk factors are continuously discovered and more than ten risk genes for late-onset AD have been identified [34-38]. In 2013, Lambert *et. al* [39] applied Meta-analysis on over 74,000 individuals and identified 11 new loci that appear to contribute to the risk of AD (the Manhattan plot shown in Figure 1). In 2014, Sage Bionetworks held a competition called The Dialogue for Reverse Engineering Assessments and Methods Challenge (DREAM Challenge), which tried to use crowdsourcing to assess the ability of current methods to predict the change in cognitive examination performance based on genetic data [40]. Some of competition teams tried to use the gene list revealed by Lambert to predict the declination of AD. However, no significant contribution of genetics beyond *APOE* haplotype to predictive performance was observed by any of the competition team.

doi:10.6342/NTU202000613

The functions of candidate genes in previous studies are diverse, including the roles in the amyloid beta pathway (*APP*, *PSEN1*, *PSEN2*, *APOE*, *CLU*, *CR1*, *PICALM*), tau pathway (*APP*, *CASS4*, *FERMT2*), endocytosis (*APP*, *SORL1*, *CD33*, *CD2AP*, *PICALM*, *BIN1*), synaptic plasticity (*PSEN1*, *PSEN2*, *APOE*), immunity and inflammation (*CLU*, *CR1*, *EPHA1*, *ABCA7*, *MS4A4A*, *MS4A6E*, *CD33*, *HLA-DRB5-DRB1*, *INPP5D*, *MEF2C*, *TREM2*), lipid transport and metabolism (*APOE*), cholesterol metabolism (*ABCA7*), apoptosis (*BIN1*, *CLU*), cell migration (*PTK2B*), hippocampal synaptic function (*MEF2C*, *P2K2B*), microglial and myeloid cell function (*INPP5D*), and cytoskeletal function (*CELF1*, *NME8*, *and CASS4*). However, the results of the studies still focused on the single single-nucleotide polymorphisms (SNP) associated to the disease. In this study, we want to figure out potential interactions between SNPs and environmental factors that are associated with AD and use these factors to predict the declination of AD.

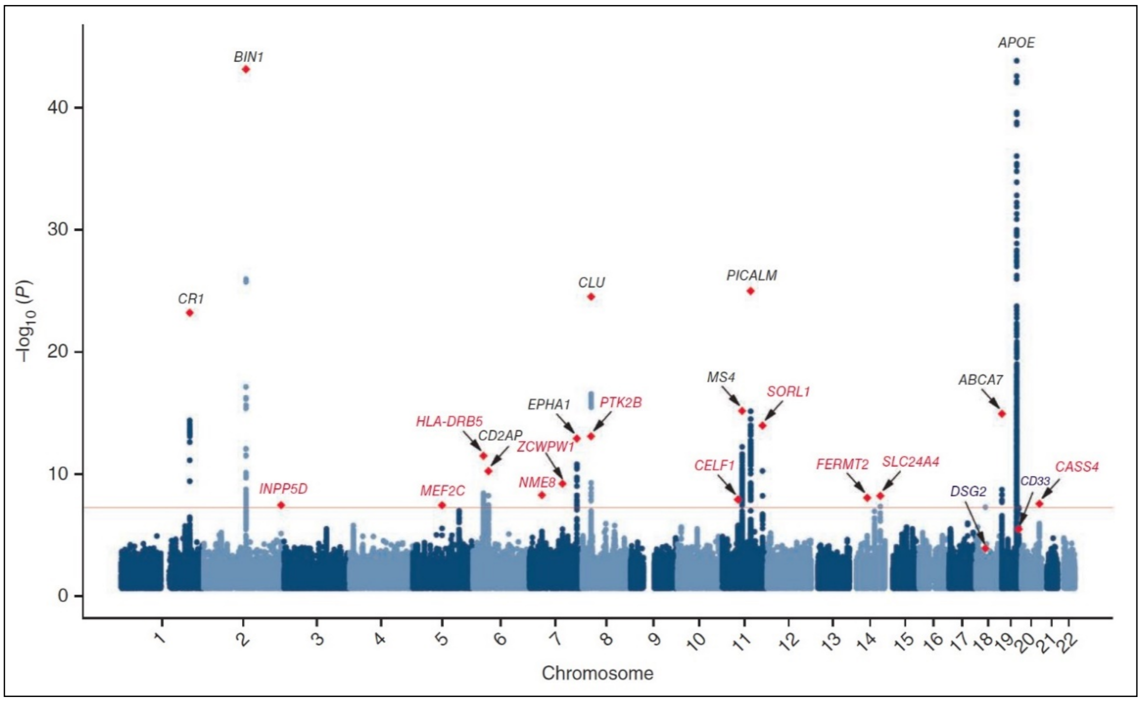

Figure 1. Manhattan plot of genome-wide association on Alzheimer's disease [39]. There are 17,008 cases and 37,154 controls. The threshold for genome-wide significance ($P < 5 \times 10^{-8}$) is indicated by the red line. Genes previously identified by GWAS are shown in black, and newly associated genes are shown in red. Red diamonds represent SNP with the smallest P values in the overall analysis.

doi:10.6342/NTU202000613

2.2. Association of genetic variants and phenotypes

Fifty years ago, doctors noticed that some of the patients who received the anesthetic succinylcholine awoke normally but remained temporarily paralyzed and unable to breath. Later scientists knew the answer: it is because that roughly 1 in 3,500 people carries particular genetic variants that affect the metabolism of succinylcholine and let them suffer the uncomfortable side effects. Moreover, it's the first link between the genetic variation and individual's drug response [41]. Relying on Genome-Wide Association Studies (GWAS), this kind of link between genetic variants and phenotypes is no longer impossible to be detected. It is a univariate examination of a genome-wide set of genetic variants in different groups to see if any variant associated with the phenotype [42]. The first GWAS was published in 2002 [43], but only after three years later, the first successful GWAS regarding age-related macular degeneration (AMD) was published [44]. It investigated the association of 105,980 SNPs with AMD in 96 cases and 50 control individuals and the result showed that the SNPs in the complement factor H (*CFH*) gene, including a non-synonymous SNP, were significantly associated with AMD. Furthermore, these findings together with biological validations. It demonstrated and convinced scientists that GWAS has its potential to identify genetic variants associated with many types of phenotypes such as diseases, cancers and drugs. Up to 2019, there are hundreds or thousands of individuals being tested in a typical GWAS, and over 210,498 variant-disease associations between 117,337 SNPs and 10,358 phenotypes were discovered [45]. However, the univariate approach analysis of GWAS limited to the success in monogenic phenotypes—in other words, it underestimates the interaction of the genetic variants, also known as epistasis, which might largely contribute to the phenotype [46].

doi:10.6342/NTU202000613

Table 1. Epistasis—the interaction of two gene loci determine the colours of Labradors. Where B-E- are black colour; bbE- are chocolate; --ee are yellow and the symbol - means it could be any allele type.

	BE	Be	bE	be
BE	BBEE (Black)	BBEe (Black)	BbEE (Black)	BbEe (Black)
Be	BBEe (Black)	bbee (Yellow)	BbEe (Black)	bbee (Yellow)
bE	BbEE (Black)	BbEe (Black)	bbEE (Chocolate)	bbEe (Chocolate)
be	BbEe (Black)	bbee (Yellow)	bbEe (Chocolate)	bbee (Yellow)

2.3. Interaction of genetic variants - Epistasis

Researchers found that GWAS failed to identify the association in multigenic phenotypes. Easton and Eeles *et. al* [47] reported a review of breast, prostate, colorectal, lung and skin cancers, and it shows that a number of new susceptibility loci have been identified using the GWAS approach but the new susceptibility loci typically have very small effect sizes. It is because that GWAS analysis considers only one genetic variant at a time thus ignores underlying epistasis of each variant [48]. We can take the colours of Labrador as an example to explain what the epistasis is and how it affects the phenotype. Labrador are registered in three colours: yellow, chocolate and black, which result from the interplay between two genetic loci B (Brown) and E (Extension). These two colours are direct production and expression of two pigments, eumelanin and pheomelanin [49] (shown in Table 1). Table 1 showed the locus E affects the expression of locus B. If the locus E has any dominance E, the locus B is the dominance and it determines the colour to be dominant black or recessive chocolate. However, if the locus E is recessive genotype ee, the colour of the dog will be yellow no matter what genotype of the locus B is [50]. The phenomenon that the phenotypic effect of locus B is masked by the genotype of locus E is called epistasis. The original concept of epistasis was introduced over a century ago [51], but now in recent literatures, it can be generally characterized as interactions between the genetic variants [52]. In the previous example, we introduced the concept of

doi:10.6342/NTU202000613

epistasis and researchers have also demonstrated that the existence of epistasis is an important factor contributing to the phenotypes, especially in complex diseases such as hypertension, diabetes and obesity [53]. Therefore, developing an analysis method to identify epistasis can help to provide a further understanding of associations between genetic variants and phenotypes [54, 55], and find out the missing heritability to compensate the deficiency of traditional GWAS. In this regard, this topic has attracted a wide range of research interests in recent years [56, 57].

2.4. The main challenges of detecting epistasis

The main challenges to develop an analysis method are computational complexity and statistical power [58]. The first challenge comes from the problem of the combination nature that when more genetic variants are considered, the number of interactions increase exponentially, known as the curse of dimensionality. Based on the specification of the most popular genomic technology company Illumia, a whole-genome microarray can interrogate over 4 million markers per sample. In order to evaluate the pairwise interactions from this microarray, we need to proceed about 8×10^{12} statistical tests. Even though Marchini *et. al* [59] has demonstrated that pairwise interactions of 3×10^{5} loci is computationally possible on current computational resources, it still remains a large gap in real microarray data. The second challenge is statistical power. Since the huge number of statistical tests are conducted on a limited sample size with high-dimension interactions, many false positives will associate with the phenotype by random chances. For these reasons, several researches endeavored to develop new methods in order to solve these problems [60, 61].

doi:10.6342/NTU202000613

2.5. Statistical and linear-base algorithms

PLINK is a one of most famous and successful programs. The method of PLINK was first introduced in 2007 [62], and Chang improved it as a full-featured GWAS software and released it in 2017 [63]. PLINK not only included traditional GWAS, but also included epistasis analysis that was designed for the detection of SNP-SNP pairwise interactions in large-scale case-control association studies. The epistasis testing of PLINK is based on the Z-score for the difference in the odds ratio of SNP-SNP association between cases and controls. The procedure of epistasis testing are constructing an allelic test of a single locus, twice collapsing three genotype categories into two allele categories (shown as Figure 2). Based on the 2×2 table, the odds ratio between loci A and B and its standard error are calculated in the standard manner. Next, the test for epistasis is the difference of the two odds ratios (shown as Equation 1).

$$\text{Z score} = \big(\log(\text{Cases}) - \log(\text{Control})\big) \,/\, \text{sqrt}\big(\text{SE}(\text{Cases}) - \text{SE}(\text{Control})\big)$$

(Equation 1)

This method was developed for dichotomous classification problem, but some researchers like Schüpbach *et. al* who extended this method to handle continuous phenotypes called FastEpistasis [64]. However, the authors mentioned that this method is used for quick screening the large-scale studies and it will cause many type I errors without given a proper threshold. Therefore, the authors also recommended using a more standard logistic regression test such as BOOST to evaluate the epistasis.

	BB	Bb	Bb
AA	a	b	C
Aa	d	e	F
aa	g	h	I

→

	B	b
AA	2a+b	2c+b
Aa	2d+e	2f+e
aa	2g+h	2i+h

→

	B	b
A	4a+2b+2d+e	4c+2b+2f+e
a	4g+2h+2d+e	4i+2h+2f+e

Figure 2. The procedure to obtain 2×2 two allelic categories table in PLINK.

doi:10.6342/NTU202000613

BOolean Operation-based Screening and Testing (BOOST) was developed by Xiang [65], which is a simple and exhaustive method. BOOST discovers gene-gene interactions based on the logistic regression model, and it will calculate the maximum log-likelihood between the main effect model and the full model that contains the interaction term (as Equation 2 and Equation 3), respectively.

$$\log\frac{P(Y=1 \mid X_p=i,\ X_q=j)}{P(Y=2 \mid X_p=i,\ X_q=j)}=\beta_0+\beta_i^{X_p}+\beta_j^{X_q}$$

(Equation 2. The logistic regression model with only main effect)

$$\log\frac{P(Y=1 \mid X_p=i,\ X_q=j)}{P(Y=2 \mid X_p=i,\ X_q=j)}=\beta_0+\beta_i^{X_p}+\beta_j^{X_q}+\beta_{ij}^{X_pX_q}$$

(Equation 3. The logistic regression model with main effect term and interaction term)

According to the likelihood ratio test, interaction effects are defined as the difference of their deviance. Xiang also claimed that applying BOOST on the Wellcome Trust Case Control Consortium (WTCCC) dataset only took 60 hours to completely evaluate all pairs of 360,000 SNPs on a 3.0 GHz desktop. Nevertheless, this number of SNPs is still far below the real data. In order to speed up on the performance of BOOST, Xiang developed a new version named GBOOST, which is a GPU-implementation of BOOST based on the CUDA technology from Nvidia. For the same interaction analysis on the WTCCC dataset, GBOOST can complete the analysis in 1.34 hours on a desktop computer equipped with standard Nvidia GeForce GTX 285 display cards. Despite the dramatically speedup of computational performance, there is still lack of a remarkable biological meaning to prove the success of these methodologies. In 2016, Murk used FastEpistasis and BOOST to search epistasis on a huge dataset called Genetic Epidemiology Research on Adult Health and Aging (GERA) study that included 78,486 subjects, but still failed to detect a significant, replicable interaction after exhaustively searching through 45 billion possible interactions in each of 10 different complex diseases [66].

doi:10.6342/NTU202000613

Previous researches showed that the parametric linear statistical model plays an important in this problem because it has solid theoretical foundation and easy to implement, but linear-based model still has limitations [67]. The first issue is that the parametric linear model generally assumes interaction effects as additive factors, which independently exhibited with marginal effects. This assumption implicit the genetic architecture of relations between genotype and phenotype is simple. However, biological systems are driven by complex biomolecular interactions, the genetic architecture is more likely a complex phenomenon such as gene-gene interaction or the example of epistasis we mentioned above [49], while the phenotype may also be affected by environment factors [68]. Moreover, Wahlsten *et. al* [69] also said that the linear model is more powerful to detect marginal effects than interaction effects. In this regard, a linear-based model would fail to detect the interaction effects in the absence of marginal effects. Therefore, we need a proper model to fit the realistic genetic architecture. The second issue is that when we want to illustrate non-linear mapping from genotypes to phenotypes, we will multiply SNPs to generate interaction terms. As the number of the phenotype goes up exponentially as each SNP is added into the model, it will cause problems when estimating parameters of the linear model, due to the empty cells that mean no data are observed for individual genotype interactions [58].

2.6. Machine learning approaches for detecting epistasis

The limitations of the linear model and the successes of machine learning motivated the researchers to address the problem of epistasis with new methodologies [70]. The first advantage of machine learning approaches is they need relative fewer assumptions and they just let data to tell us which one is the best model of various learning algorithms rather than enforcing the data to fit the preconceived model. The second advantage of

doi:10.6342/NTU202000613

machine learning is some learning algorithms can properly fit the non-linear relations of genetic architectures such as random forests [71, 72] or even more researchers designed tailor-made algorithms such as Multifactor Dimensionality Reduction (MDR) [67] and RELIEF [73] for solving this problem.

The method Random Forests was applied successfully on genetic data in various studies [74, 75] and it could either be used in dichotomous classification or continuous regression problems. Random forests consist of multiple sets of decision trees. Each decision tree is trained using a bootstrap procedure [76] (shown as Figure 3). Then, the result would be estimated based on aggregated voting over all trees in the forest. The first step of the procedure is choosing a training set by selecting N samples from the data and the rest of the samples are regarded as the testing set. In the second step, m attributes would be randomly selected from M attributes in the data at each node in the tree. The third step is choosing the best split at that node from among the m attributes based on such as Gini index [77, 78] or information gain criteria [79]. The final step iterates the second to third steps until the tree is fully grown and the out-of-bag testing set would be used to estimate the performance of each tree. A forest would be generated by repeating this procedure. The nonlinear fashion of random forests fits the SNP data properly and is

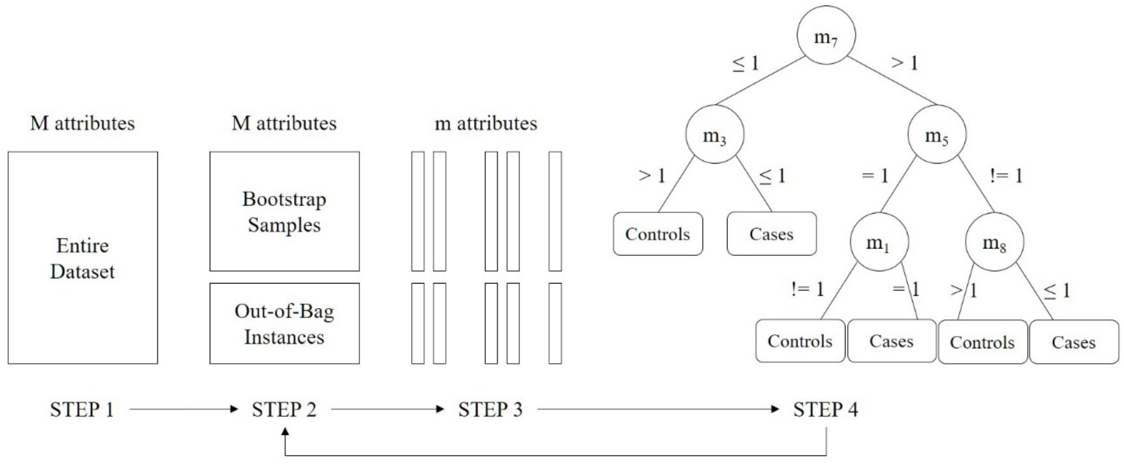

Figure 3. The procedure of random forest algorithm.

doi:10.6342/NTU202000613

able to detect small effects because the small effect can be found by splitting the tree to fit subsets of data. It has been shown that random forests can substantially be more efficient than standard statistical methods in ranking the true disease-associated SNPs in order to detect epistasis [80]. However, when the amount of the attributes is large, the computing time increase dramatically on finding the best splitting attribute m. Thus, researches recommended using filtering algorithms such as MDR [67] and RELIEF [73] or splitting the genotype data to avoid this problem [81].

MDR was developed by Moore in 2002 [67], which is a non-parametric and genetic model-free data mining and machine learning strategy for identifying epistasis, including the interactions of genetic variants and environmental factors [60]. The summary of MDR is as shown in Figure 4. The first step of MDR is to select M factors from the pool of all the genetic and environmental factors. The second step is to generate the possible combinations and to calculate the case-control differences for each combination. Taking two genetic variants as an example, each locus has three genotypes, there are nine combinations in total. In the third step, each combination would be identified as a high or low risk cell. Then, all of the high-risk cells would be grouped into one new factor and the low-risk factors would be grouped into another new factor. This procedure demonstrated that MDR is also a good way to reduce the dimension of attributes. The

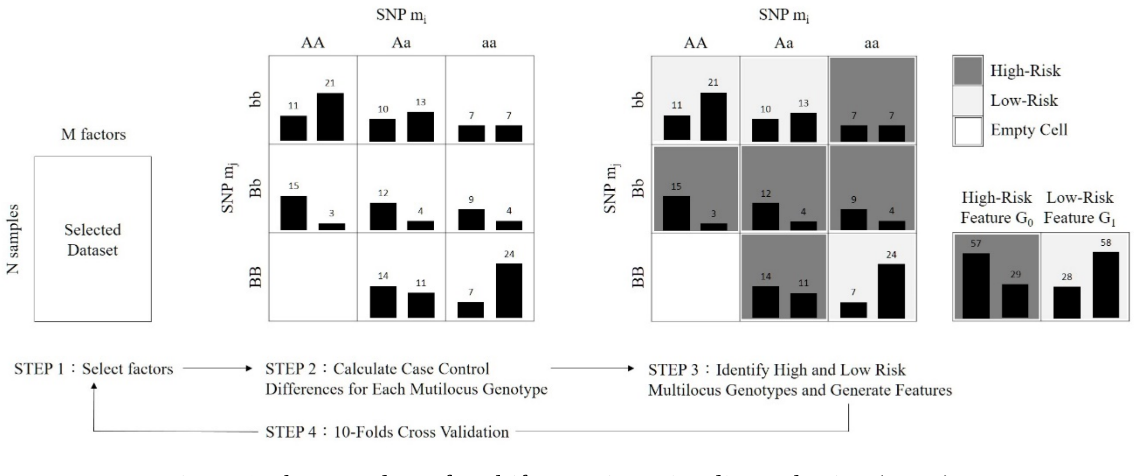

Figure 4. The procedure of Multifactor Dimensionality Reduction (MDR).

doi:10.6342/NTU202000613

new factors generated in third step can further be used to predict the phenotype. The prediction errors would be estimated in the fourth step. Based on this original concept, many improvements and extensions of MDR have been released [82-84] and this method has also been confirmed in numerous simulation studies.

Kira and Rendell first introduced RELIEF in 1992 [85], which estimates the quality of each attribute through a nearest algorithm that calculates the differences between the randomly selected instances and its neighbor instances. Figure 5 is an example of dichotomous classification problem. In the beginning of the algorithm, RELIEF declares an empty array W to memorize the weights of each attribute. Next, RELIEF randomly selects an instance m times. In each selection, RELIEF estimates the weight of each attribute (A) based on the difference between the randomly selects instance (R) and its nearest neighbor, which can be defined as two types including the same class (nearest hit, H) and the different class (nearest miss, M). After m repeats, the algorithm outputs the array W with weights for each attribute ranging from –1(worst) to +1(best). RELIEF can act as a filtering algorithm based on the weight array and it is also able to detect epistasis. This is because it considers the entire vector of values across all attributes when it selects the nearest neighbor. However, this feature is also a disadvantage because the entire vector might cause many noisy and potential false positives [58].

```
set all weights W[A] := 0.0;
for i := 1 to m do
    begin
        randomly select an instances R;
        find nearest hit H and nearest miss M;
            for A := 1 to all_attributes do
                W[A] := W[A] – diff(A,R,H)/m + diff(A,R,M)/m;
    end;
```

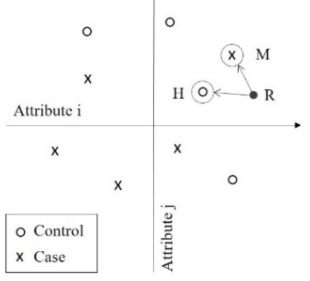

Figure 5. The pseudocode and concept of RELIEF algorithm.

doi:10.6342/NTU202000613

3. Materials and Methods

3.1. Alzheimer’s Disease Neuroimaging Initiative (ADNI) dataset

In total, our study consists of 767 participants were healthy elderly and mild cognitive impairment (MCI) and Alzheimer’s disease patients from the Alzheimer’s Disease Neuroimaging Initiative (ADNI). All participants followed the study protocols approved by ADNI and provided signed informed consent. The description and preprocessing of data are described in detail in the following sections.

ADNI is a global study effort that actively supports the investigation and development of treatments that slow or stop the progression of AD. This multicenter, longitudinal study has three phases: ADNI-1, ADNI-GO and ADNI-2 which collects clinical assessment and measures, imaging that includes magnetic resonance imaging (MRI) and Positron emission tomography (PET), genetic and Biospecimen biomarkers through the process of control normal aging (CN) to early mild cognitive impairment (EMCI), too late mild cognitive impairment (LMCI), to AD. The 767 ADNI participants that we used consists of 241 CN, 130 EMCI, 273 LMCI and 123 AD participants. We accessed two types of features in this study were clinical and genetic data. The clinical assessment is the Mini-Mental State Examination (MMSE) at baseline and the 24-month follow-up visit. Other clinical measures included diagnosis at baseline, age at baseline, years of education at baseline, gender, APOE genotype. The genetic data were genotyped on the Illumina Human610-Quad BeadChip in ADNI-1 and Illumina HumanOmniExpress BeadChip in ADNI GO/2. The multidimensional scaling analysis being applied by PLINK using HAPMAP3 to ensure that sample within the European HapMap populations. Then, the data were imputed to 1,000 genomes haplotypes phase one integrated variant set. After imputation, there were 12,809,667 genotyped features in total.

doi:10.6342/NTU202000613

3.2. The architecture of GenEpi

The architecture of GenEpi is shown in Figure 6. GenEpi is designed to group SNPs by a set of loci in the gnome. For examples, a locus could be a gene. In other words, we use gene boundaries to group SNPs. A locus can be generalized to any particular regions in the genome, e.g. promoters, enhancers, etc. GenEpi first considers the genetic variants within a particular region as features in the first stage, because it is believed that SNPs within a functional region might have a higher chance to interact with each other and to influence molecular functions. The idea of within-gene epistasis analysis followed by cross-gene analysis is not new, which has also been used in previous studies [86-89]. Differently, GenEpi adopts two-element combinatorial encoding when producing

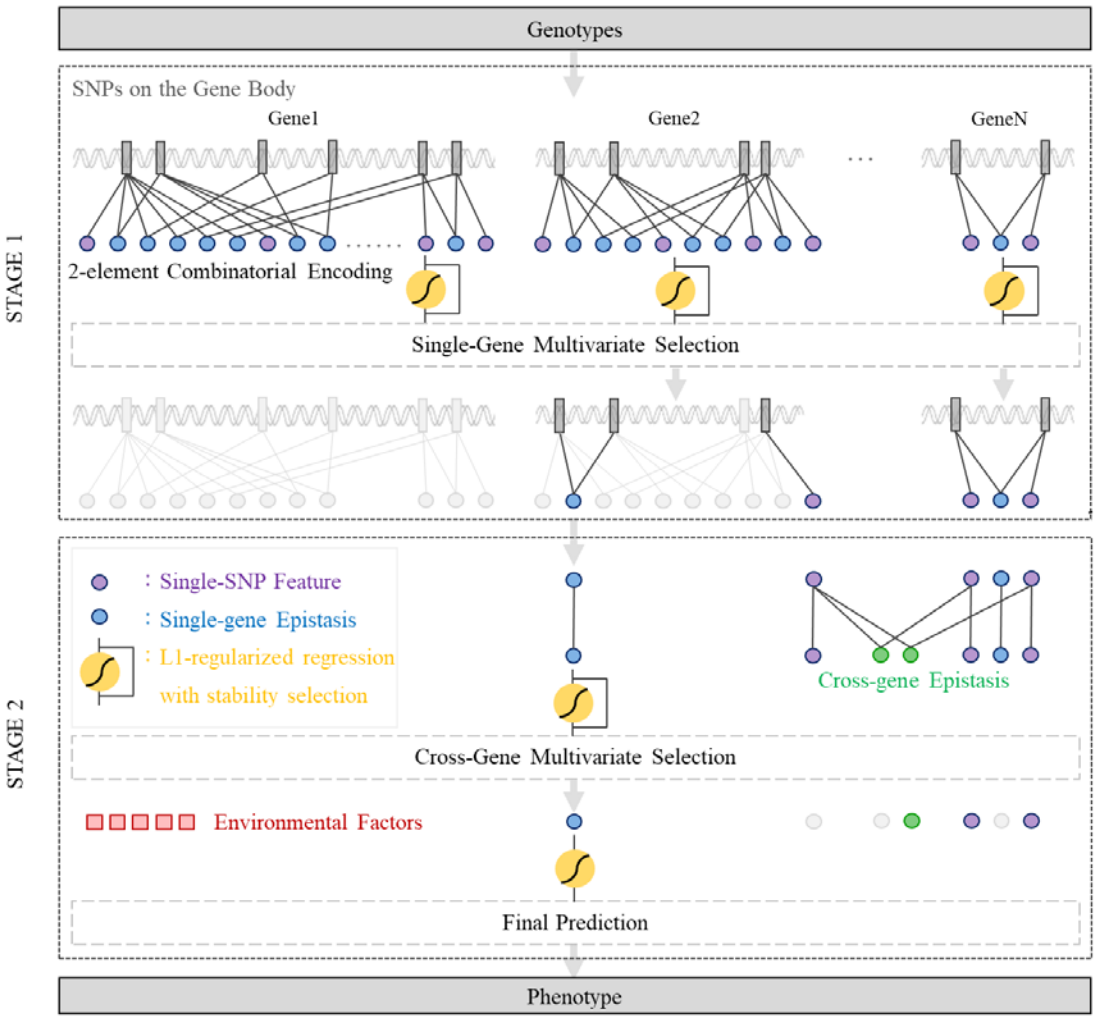

Figure 6. The architecture of GenEpi.

doi:10.6342/NTU202000613

features and models them by L1-regularized regression with stability selection, which will be explained in Section 3.5. In the first stage (STAGE 1) of GenEpi, the genotype features from each single gene will be combinatorically encoded and modeled independently by L1-regularized regression with stability selection. In this way, we can estimate the prediction performance of each gene and detect within-gene epistasis with a low false positive rate. In the second stage (STAGE 2), both of the individual SNP and the within-gene epistasis features selected by STAGE 1 are pooled together to generate cross-gene epistasis features, and modeled again by L1-regularized regression with stability selection as STAGE 1. Finally, the user can combine the selected genetic features with environmental factors such as clinical features to build the final prediction models. In addition to the main procedures, two pre-processing steps are also implemented in GenEpi: retrieving the gene information from public databases and and reducing the dimensionality of the features using linkage disequilibrium (LD). In the end, we released a Python package that implements GenEpi. The details of these steps and the GenEpi method will be described in the following sections.

3.3. University of California Santa Cruz (UCSC) database

To obtain the gene information such as official gene symbols and genomic coordinates, we retrieved kgXref and knownGene data table from the UCSC human genome annotation database [90, 91]. The version of the database we used is the Feb. 2009 assembly of the human genome hg19, GRCh37 Genome Reference Consortium Human Reference 37. The two data tables were merged in order to generate a local database containing the gene symbols as well as the genomic coordinates of each gene. The in-house script we built could update this local database automatically. It is noted that there are many different categories of genes in the RefSeq database. In this study, we only

doi:10.6342/NTU202000613

focused on the mRNA and non-coding RNA (22,376 genes in total). The selected transcripts were projected on the genomic coordinates and the coordinates of corresponding genes were determined based on the leftmost and rightmost positions of the corresponding transcripts. Moreover, to discover the factors that might affect the transcription of genes, we also retained the promoter region of each gene. In genetics, the promoter region is a segment of DNA that initiates the transcription of a particular gene. Promoters are located near the transcription start sites of genes, on the upstream of the same DNA strand (towards the 5' region of the sense strand of the transcript). In general, a promoter region can be 100-1000 base pairs long. In this study, we extracted 1,000 nucleotides on the upstream of the starting position of each gene as the promoter region.

3.4. Estimation of linkage disequilibrium

Linkage disequilibrium (LD) was first used in 1960 by [92] and refers to the non-random association of alleles at different loci in a given population. Loci are said to be in linkage disequilibrium when the frequency of association of their different alleles is higher or lower than what would be expected if the loci were independent and associated randomly. Linkage disequilibrium or equilibrium depends on the occurrence of recombination, which is facilitated by chromosomal crossover during meiosis and that would result in descendants having different combinations of genes and occasionally produce new allele (shown in Figure 7). We can consider recombination as a shuffling procedure, and in a younger population such as European, which has fewer shuffling times so their SNP features may have more dependency, resulting in larger LD blocks. In machine learning, we can take the advantage of LD to reduce the dimension of SNP features. In this regard, we followed the study of Lewontin [93] to estimate LD using Equation S4-6 to reduce

doi:10.6342/NTU202000613

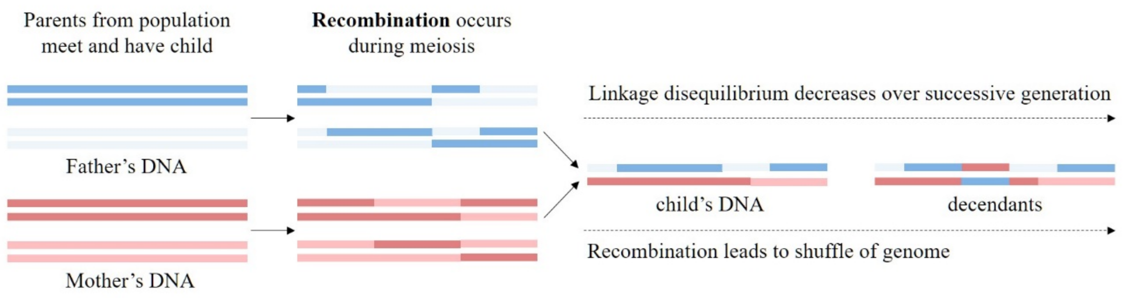
Parents from population meet and have child
Recombination occurs during meiosis
Father's DNA
Mother's DNA
Linkage disequilibrium decreases over successive generation
child's DNA
decendants
Recombination leads to shuffle of genome

Figure 7. Illustration of linkage disequilibrium.

the redundant SNP features. Equation 4 calculates the coefficient of linkage disequilibrium *D*, where p_A is the proportion of chromosomes with the allele type A at one locus, p_B is the proportion with the allele type B at another locus, and p_{AB} is the proportion of chromosomes with both A and B on the same chromosome. In genotyped data, we do not have phased information to calculate p_{AB}, therefore we used the Hill's Expectation-Maximization (EM) algorithm approach [94] to predict the haplotype and estimate p_{AB}..

$$D = p_{AB} - p_A p_B$$

(Equation 4. The formula of the coefficient of linkage disequilibrium *D*)

$$D' = D \ / \ D_{\min} \ \text{, where}$$

$$D_{\min} = \begin{cases} \max\{-(p_A p_B), -(1-p_A)(1-p_B)\} & \text{when } D < 0 \\ \min\{p_A(1-p_B), (1-p_A)p_B\} & \text{when } D > 0 \end{cases}$$

(Equation 5. The formulas of normalized *D*)

Equation 6 is the formula of correlation coefficient r^2, which is an alternative to *D'*, referring to the correlation coefficient between pairs of loci. We used D' >0.9 and r^2>0.9 as the criteria to group highly dependent SNP features as blocks. In each block, we chose the features with the largest minor allele frequency to represent other features in the same block. It is important to look at the SNPs falling in the same LD blocks with the SNPs

$$r^2 = \frac{D^2}{p_A \ (1-p_A) \quad p_B \ (1-p_B)}$$

(Equation 6)

doi:10.6342/NTU202000613

discovered by GenEpi. Some true interactions might be skipped owing to some strong signals provided by the SNPs in the same LD block.

3.5. Discovery of within-gene epistasis

The main objective of the first stage in GenEpi is to select candidate features from each gene. In order to extract SNP features for a gene, we used the start and end positions of each gene from the local UCSC database to split the SNP features after dimension reduction. Since there are 22,376 genes in the UCSC database, we obtained 22,376 subsets of the SNP features. In each subset, a SNP feature with the alleles 'A' and 'a' could have three possible genotypes, AA, Aa and aa, which are used to refer to the pairs of alleles. The pairs of alleles are subsequently separated into three binary features using one-hot encoding. In order to evaluate epistasis, we generated interacting features by crossing each pair of genotype features. Considering the false positive rate and computational complexity, we only focused on pairwise interactions of epistasis throughout this study. We defined the interaction between two SNPs in Equation 7. In Equation 7, $\alpha_1 SNP_1 + \alpha_2 SNP_2$ stand for the additive interactions and $\alpha_{int(1,2)} SNP_1 \otimes SNP_2$ represents the synergistic interactions that contain nine terms.

$$y = \alpha_0 + \sum_{m \in \{AA,Aa,aa\}} \alpha_{1,m} SNP_{1,m} + \sum_{m \in \{AA,Aa,aa\}} \alpha_{2,m} SNP_{2,m} + \sum_{m,n \in \{AA,Aa,aa\}} \alpha_{1,m,2,n} SNP_{1,m} SNP_{2,n}$$

$$= \alpha_0 + \alpha_1 SNP_1 + \alpha_2 SNP_2 + \alpha_{int(1,2)} SNP_1 \otimes SNP_2$$

(Equation 7)

Before modeling each subset of genotype features, two criteria were adopted to exclude low quality data. The first criterion is that the genotype frequency of a feature should exceed 5%, where the genotype frequency means the proportion of a genotype among the total samples in the dataset. The second criterion is regarding the association between the feature and the phenotype. If the phenotype is dichotomous or categorical, the p-value of

doi:10.6342/NTU202000613

χ^2 test should be smaller than 0.01. If the phenotype is a real number, we used F-test to estimate the correlation between the feature and the phenotype, and the p-value should be smaller than 0.01. In the end, a gene may have multiple SNPs. The general form of the linear model for a gene with k SNPs is defined as Equation 8, which is termed as two-element combinatorial encoding.

$$y = \alpha_0 + \sum_{i=1}^{k} \alpha_i SNP_i + \sum_{i=1}^{k} \sum_{j=1 \cap j \neq i}^{k} \alpha_{int(i,j)} SNP_i \otimes SNP_j$$

(Equation 8)

We conducted L1-regularized regression [95] with stability selection [96] for modeling each gene. The sparsity of the L1-regularized model prefers solutions with a smaller number of features, which effectively reduces the number of features. As in Equation 9, L1-regularized regression uses an additional regularization term $\lambda\|\boldsymbol{\alpha}\|_\mathbf{1}$ to restrict the weight of each feature by shrinking some of them to 0 so that the non-zero remainders can represent the exact set of true features when given a proper λ. In Equation 9, we have the vector $\mathbf{SNP} = (SNP_1, \dots, SNP_i, \dots, SNP_k, SNP_1 \otimes SNP_2, \dots, SNP_i \otimes SNP_j, \dots, SNP_{k-1} \otimes SNP_k)$, the corresponding coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_i, \dots, \alpha_k, \alpha_{int(1,2)}, \dots, \alpha_{int(i,j)}, \dots, \alpha_{int(k-1,k)})$, the target y_l takes the values $\{-1, 1\}$ at sample l and c is a constant to be determined during modeling.

$$\widehat{\boldsymbol{\alpha}}^\lambda = \min_{\boldsymbol{\alpha},\boldsymbol{c}} \sum_{\boldsymbol{l}=\mathbf{1}}^{\boldsymbol{n}} \log\left(\exp\left(-\boldsymbol{y_l} \times \left(\mathbf{SNP}_{\boldsymbol{l}}^{\mathbf{T}} \boldsymbol{\alpha} + \boldsymbol{c}\right)\right) + \mathbf{1}\right) + \lambda\|\boldsymbol{\alpha}\|_\mathbf{1}$$

(Equation 9)

, where n stands for the number of samples. It should be noticed that, if the features are conditional dependent, the solution of these equations will not be unique. It would lose generality to determine the proper amount of λ when we only consider a possible solution of weight vector $\boldsymbol{\alpha}$. Resampling is an intuitive technique to increase the generality, which

doi:10.6342/NTU202000613

can largely reduce the false positive rate. Here, we used stability selection [96] to tackle this problem. Stability selection works by resampling and remodeling the training set hundreds of times, followed by picking out the features that are repeatedly selected across randomization. In this study, we executed this randomization 500 times, and the features selected by stability selection would be retained for the next stage.

3.6. Discovery of cross-gene epistasis

In the second stage, we used the features selected by STAGE 1 to generate cross-gene epistasis features. To avoid missing any possible association between genotype features and phenotype. In the default setting of the GenEpi package, we include all the genes with non-zero F1 score or non-zero correlation to go into the next stage. Then we applied the same selection procedure described in Section 3.5 to find the cross-gene epistasis that are associated with the phenotype. The procedures were slightly modified here. Since we only focused on pairwise interactions, instead of using the entire features we selected in STAGE 1, we only used single-SNP features to generate cross-gene epistasis features. Also, we used the genotype frequency and the p-values of χ^2 test or F-test to control the quality of features and to avoid overfitting. Nevertheless, the p-value of each feature in this stage should be smaller than 10^{-5}. All of the features from different genes would be merged for modeling cross-gene epistasis. We conducted L1-regularized regression for modeling, and the stability selection were used once again to select the final genotype feature set. Since the phenotype may also be affected by environmental factors, after determining the final set of genotype features, the user can include the environmental factors such as clinical assessments for constructing the final model. Subsequently, the final model was evaluated through a process called double cross validation (CV). In the external loop of double CV, all the instances were divided into two subsets to serve as

doi:10.6342/NTU202000613

training and independent test sets. In this study, we used 2-fold CV and leave-one-out CV (LOO CV) in external loop for evaluation. In the internal loop, we also used 2-fold CV for model selection.

doi:10.6342/NTU202000613

4. Results

This study compared GenEpi with several commonly used algorithms for detecting epistasis, including FastEpistasis, BOOST and ReliefF. The simulation data demonstrated that GenEpi outperforms the other methods in ranking the true epistasis as the top one. As real data is concerned, the results suggest that the epistasis selected by GenEpi has the best predictive power for diagnosis of AD. The proposed model of predicting AD contains three clinical features, 14 genetic features, including 24 SNPs from 12 genes that contain the well-known causal gene, APOE. The 2-fold cross validation (CV) and leave-one-out CV (LOO CV) accuracy of this model are 0.829 and 0.832, respectively. On the other hand, for predicting progression, the proposed model contains eight clinical features, four genetic features, including seven SNPs from six genes, where all of the four genetic features are cross-gene epistasis with significant p-values ($< 10^{-11}$). The average of Pearson and Spearman correlation of 2-fold CV and LOO CV are 0.52 and 0.53, respectively. The results demonstrated that GenEpi has the ability to detect the epistasis associated with the phenotype effectively and efficiently. The released package can be generalized to largely facilitate the studies of many complex diseases in the near future.

We will demonstrate our experiments in following four parts of this section. In the first part, we applied GenEpi and other algorithms for detecting epistasis, including FastEpistasis [18], BOOST [19] and ReliefF [23, 97] on simulation data for validation and comparison. In the second part, we applied GenEpi on the ADNI dataset to categorize each sample as control subjects or AD patients, evaluated by precision, recall, accuracy and F1 score ($2 \times$ (precision $\times$ recall) / (precision + recall)). In the third part, we used the same dataset to predict the progression of AD, which is being defined as the MMSE difference between the baseline and the follow-up after 24 months, and the performance was assessed using Pearson correlation and Spearman correlation between the predicting

doi:10.6342/NTU202000613

results and the MMSE differences. In the final part, we compared GenEpi with other algorithms on the ADNI dataset in terms of computing time and prediction performance on real data.

4.1. Experiments on simulation data

We applied different algorithms on simulation data for validation and comparison. All of the simulation datasets are generated by the simulator GAMETES [98], which is publicly accessible on the web https://popmodels.cancercontrol.cancer.gov/gsr/packages/gametes/ We designed two types of simulation datasets: basic and complex models. The 'Model 1', 'Model 2' and 'Model 3' are simulation datasets with the basic model, which means that each dataset contained only one epistasis consisting of a SNP pair. All of the basic-model datasets are in the same setting as follows: #individuals = 2000, case/control ratio = 1, #SNPs = 100, #replicates = 100, minor allele frequency of target SNPs = 0.2, and heritability = 0.2. The complex model means one dataset contains multiple epistasis from different SNP pairs. Here, the 'Combined Model 1+2+3' is a complex model dataset containing three epistasis from the previous three basic models.

Figure 8 provided the results of these four simulation datasets. Figure 8(a) shows that the ranking of the target epistasis reported by GenEpi in the 100 replicates of each basic-model dataset are always ranked as the first. In contrast, for FastEpistasis and BOOST, the medians of the ranking of the target epistasis among the 100 runs of simulation are one but the averages are not. The number of failures of FastEpistasis and BOOST in 100 replicates of three basic models are 6, 1, 16 and 5, 1, 14, respectively. For the result of the complex model dataset in Figure 8(b), the superiority of GenEpi over other algorithms is more obvious. In the 100 runs of simulation, GenEpi reported the three target epistasis as the top three important features every time. In contrast, FastEpistasis

doi:10.6342/NTU202000613

and Boost failed to report the three target epistasis as the top three important features consistently. When ReliefF was compared, since the Python package scikit-rebate [97] that we used for implementing ReliefF only reports the importance of individual SNPs instead of the scores for epistasis (SNP pairs), we listed the medians of ReliefF's ranking for each SNP in the target epistasis in Table 2. Table 2 reveals that ReliefF can detect the SNPs in the target epistasis in the basic models, but failed to report the three target epistasis as the top three important features in the complex model dataset.

The superiority of GenEpi is owing to the proposed two-element combinatorial encoding of the genotype features and the L1-regularized regression with stability selection. In contrast with other statistical algorithms such as FastEpisasis and BOOST, which only evaluate the epistasis of a SNP pair one at a time, GenEpi considers

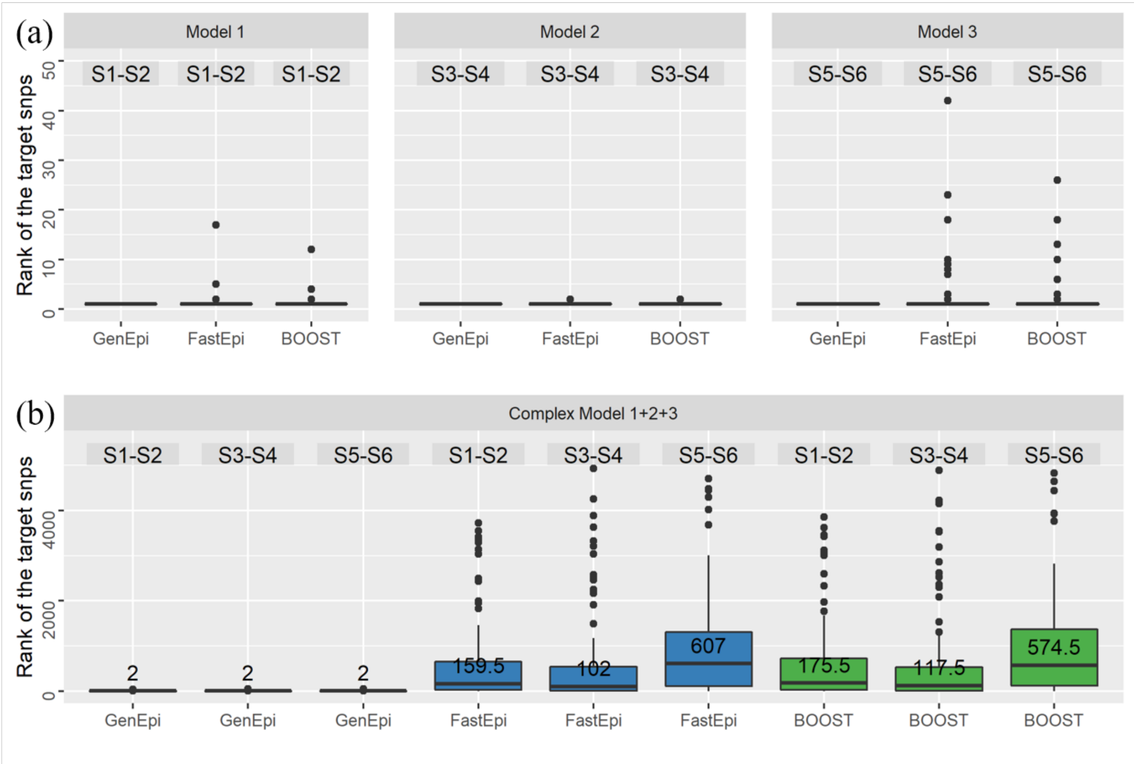

Figure 8. The boxplot for the rank of the target epistasis in different algorithms. (a) The results of three basic-model datasets with one epistasis consisting of a SNP pair. (b) The result of the complex-model dataset, which contained three epistasis. The 'S1-S2' means the epistasis between SNP 1 and SNP 2 and so on. The values on the boxplot are the medians of the rank of the target epistasis among the 100 runs of simulation.

doi:10.6342/NTU202000613

Table 2. The medians of the rank of the SNPs in the target epistasis for ReliefF.

	SNP 1	SNP 2	SNP 3	SNP 4	SNP 5	SNP 6
Basic Model	1	2	1	2	1	2
Complex Model	7	8.5	9.5	11.5	11	19.5

interactions between combinatorial features by multivariate models. Moreover, the false positives among the epistasis can be filtered out by resampling and remodeling the dataset hundreds of times. To evaluate the effect of stability selection, we applied both L1-regularized regression with and without stability selection on the complex model dataset to compare the number of false positives, which is defined as the number of non-target epistasis in the final output of GenEpi. As shown in Figure 9, stability selection can reduce the mean false positive rate and minimize the variance of false positive rate as well.

4.2. Classifying AD patients

In predicting control subjects or AD patients, we applied GenEpi on the 364 samples with CN (as control) or AD. After dimensionality reduction, 12,102,888 out of the 12,809,667 SNPs in the ADNI dataset were retained, and 4,916,249 of them are located in 20,206 genes. In the step of selecting single-gene epistasis, there are 34,689 genetic features selected and 765 of them are single SNP features, while the other 33,924 are epistasis features within genes. The final model contained 3 clinical features (Age, Gender, and Education) and 14 genetic features, including 24 SNPs from 12 genes. These

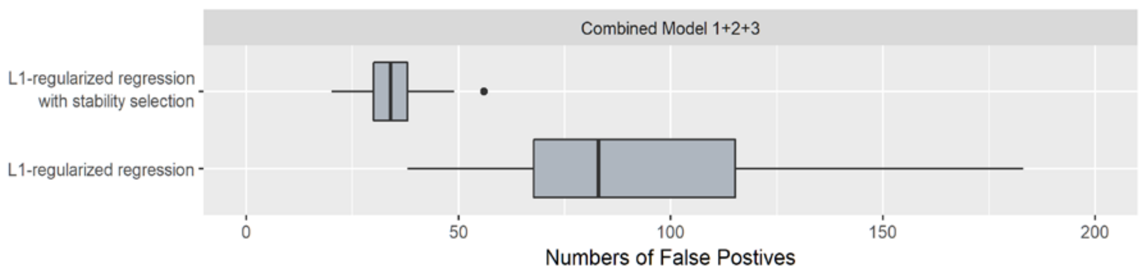

Figure 9. The boxplot of false positives in L1-regularized regression with and without stability selection.

doi:10.6342/NTU202000613

Table 3. The score of different models in predicting control subjects or AD patients.

	Precision	Recall	F1 Score	Accuracy
Training	0.9633	0.8537	0.9052	0.9396
2-fold CV	0.7748	0.6992	0.7350	0.8297
LOO CV	0.8100	0.6585	0.7265	0.8324

F1 Score = 2 × (Precision × Recall) / (Precision + Recall); 'Training' stands for the process of a single-loop CV; '2-fold CV' means that 2-fold CV was used in the external loop of double CV; 'LOO CV' means that LOO CV was used in the external loop of double CV.

features contained two single SNP features, 11 within-gene epistasis features and one cross-gene epistasis feature. As shown in Table 3, the 2-fold cross validation (CV) and leave-one-out CV (LOO CV) accuracy of this model are 0.83 and 0.83, respectively.

We listed the statistical significance of the selected genetic features in Table 4. The first column lists each feature by its RSID (Reference SNP cluster ID) and the genotype (denoted as RSID_genotype), the pairwise epistasis features are represented using two SNPs. The last column describes the genes where the SNPs are located according to the genomic coordinates. We used a star sign to denote the epistasis between genes. Here, only the feature (rs3130614_BB, rs41276317_AB) is cross-gene epistasis (for MICB and TOB2). The weights in the second column were extracted from the linear model we defined in Section 3.6. The signs of the weights indicate if a feature is a causal or protective genotype, which is consistent with the corresponding odds ratio. The p-value of the χ^2 test showed that these features are significantly associated with the phenotype.

We took one single-SNP and the cross-gene epistasis features as examples. Since each SNP pair has nine SNP combinations, we used the color 'light blue' to highlight the features selected by GebEpi. We considerd that a good feature can be judged by two conditions. The first one is that the difference between the amount of case and control subjects should be large, which suggests that the case and control subjects can be distinguished by this feature. The second one is that the genotype frequency of this feature

doi:10.6342/NTU202000613

Table 4. The statistical significance of genetic features selected by GenEpi in predicting AD.

Selected SNPs (RSID)	Weight	Odds Ratio	χ^2-test P-value	Genotype Frequency	Gene
rs3130614_BB, rs41276317_AB	3.16	19.23	1.42E-09	0.0742	*MICB*TOB2*
rs12095538_BB, rs2774308_AB	2.41	7.69	6.87E-07	0.0824	*SYT6*
rs12926153_AB, rs12922908_AA	1.18	4.83	6.89E-07	0.1511	*CLEC16A*
rs9652600_AB, rs12922908_AA	0.94	4.83	6.89E-07	0.1511	*CLEC16A*
rs9344977_BB, rs56148686_AB	1.94	4.32	1.14E-06	0.1813	*BACH2*
rs429358_AA	-2.01	0.17	1.73E-06	0.5962	*APOE*
rs56233035_AB, rs3678_AB	2.26	10.16	1.91E-06	0.0604	*CACNA1E*
rs11675339_AA, rs2710687_AA	2.32	3.94	3.55E-06	0.1923	*VSNL1*
rs12189429_BB, rs6881360_AA	1.36	4.34	3.65E-06	0.467	*ADAMTS12*
rs12187423_BB, rs6881360_AA	0.58	4.34	3.65E-06	0.467	*ADAMTS12*
rs10831829_BB, rs12366151_AA	3.48	9.50	4.90E-06	0.0577	*PARVA*
rs2052573_BB, rs34580133_AB	1.80	4.08	5.00E-06	0.1648	*LINC00299*
rs2421701_AB, rs200512701_AB	1.82	4.12	5.29E-06	0.1593	*TNKS2*
rs769449_AA	-1.19	0.16	8.42E-06	0.6648	*APOE*

The sign '*' between two gene symbols indicates cross-gene epistasis.

cannot be too small, which suggests that the occurrences of this feature are not only supported by a small number of samples. Figure 10(a) reveals that rs429358_AA is a

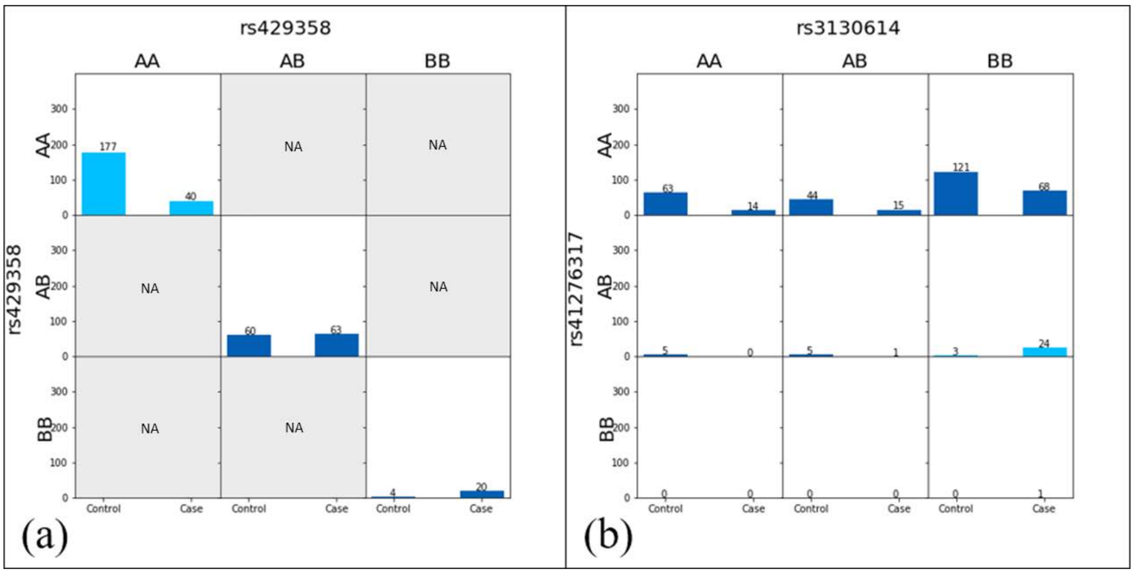

Figure 10. (a) The bar plot for the population of three allele types in rs429358. (b) The bar plot for combinations of SNPs from the cross-gene epistasis feature selected by GenEpi in predicting control subjects or AD patients. The color 'light blue' is used to highlight the features selected by GenEpi.

doi:10.6342/NTU202000613

good feature, because the number of samples in the light blue cell is about 60 percent of the whole dataset and the proportion of control versus case is 4.425:1. This feature is actually a single SNP feature located on *APOE*. Moreover, it is a well-known SNP for determining the allele type of *APOE*. Figure 10(b) showed all combinations of the SNPs rs3130614 and rs41276317. Even though the number of samples in the light blue cell is not particularly large, it can still be considered as a good feature with following explanation. By examining the cells in this figure, two of them are empty cells, one of them has only one sample, and another five cells have more control subjects than cases in the cells. Since the proportion of control subjects and cases of this dataset is 2:1, the samples seem to be randomly distributed into these five cells. In contrast, the proportion of the control subjects and cases is 1:8 in the light blue cell in the opposite, which indicates the SNP pair with such combination can be considered as a potential causal genetic variant for AD. These results show that GenEpi has the ability to identify good features.

4.3. Predicting the progression of AD

In predicting the progression of AD, we applied GenEpi on all of the samples from ADNI. For this task, the 'diagnosis at baseline' and 'MMSE at baseline' were also adopted as clinical features. After LD dimension reduction, the original 12,809,667 SNPs decreased to 12,507,343, and 5,086,639 of them are located on 20,206 genes. After selecting epistasis, there are 56,427 genetic features left, and 1302 of them are single SNP features, while the other 55,125 are epistasis features in a single gene. The final model contained eight clinical features (Age, Gender, Education, CN, EMCI, LMCI, AD, and MMSE at baseline) and four genetic features, which included seven SNPs from six genes, and all of the four genetic features are cross-gene epistasis. As shown in Table 5, the average of Pearson and Spearman correlation of 2-fold CV and LOO CV are 0.52 and 0.53,

doi:10.6342/NTU202000613

Table 5. The score of different models in predicting the progression of AD.

	Pearson correlation	Spearman correlation	Average (P+S)
Training	0.9600	0.5702	0.5732
2-fold CV	0.9385	0.4776	0.4854
LOO CV	0.9493	0.5239	0.5293

‘Pearson’ means Pearson’s Correlation; ‘Spearman’ means Spearman’s Correlation; ‘AVG(P+S)’ means the average of Pearson and Spearman correlations; ‘Training’ stands for the process of a single-loop CV; ‘2-fold CV’ means that 2-fold CV was used in the external loop of double CV; ‘LOO CV’ means that LOO CV was used in the external loop of double CV.

respectively. The weights in Table 6 were extracted from the linear model we described in Section 3.6. The signs of the weights indicate whether the feature is a causal or protective gene. We applied student’s t-test on the samples with the combination of the selected genetic features (Figure 11), and all of the p-values of t-test are significant ($< 10^{-11}$). These results show that GenEpi has the ability to identify good features in the regression task, too.

4.4. Comparison with different algorithms

In this section, we compared GenEpi with other algorithms for detecting epistasis, including FastEpistasis [18], BOOST [19] and ReliefF [23] in terms of computing time and prediction performance. We used Microsoft Azure E32 v3 as the computing resource, which contains 32 CPUs and 256 GB RAM. Since the PLINK (version 2.0) has imported

Table 6. The statistical significance of genetic features in predicting the progression of AD.

Selected SNPs (RSID)	Weight	T-test P-value	Genotype Frequency	Gene
rs2306894_BB, rs11653674_BB	-1.2045	5.53E-13	0.3025	*CLEC1A*GPR142*
rs2306894_BB, rs4078431_BB	-0.8244	4.55E-13	0.0574	*CLEC1A*PRKAG2-AS1*
rs7221462_BB, rs11721292_AA	1.3215	3.09E-11	0.8905	*IFT20*RPL32*
rs2242373_AA, rs4078431_BB	-1.2188	3.51E-13	0.0756	*MFSD6L*PRKAG2-AS1*

‘*’ sign between two gene symbols indicates cross-gene epistasis.

doi:10.6342/NTU202000613

FastEpistasis and BOOST, we used PLINK to test these two algorithms. For ReliefF, we employed a Python package called scikit-rebate [97] for implementation. Among these algorithms, only FastEpistasis can afford the computation of the whole set of SNPs. In this regard, 12,809,667 SNPs were used by FastEpistasis (Table 7). On the other hand, GenEpi only focuses on the SNPs in the gene regions. In this regard, the number of input SNPs for estimating epistasis reduced to 4,916,249. BOOST took the same subsets of SNPs as GenEpi (Table 7). When taking the same subset of SNPs as GenEpi and BOOST, ReliefF still caused memory errors. Therefore, we used the subsets of SNPs that selected by STAGE 1 of GenEpi as the input of ReliefF, which are 33,868. We selected the top 15, 30, 45 and 60 rankings from the results of these algorithms for comparing the prediction performance, and used L1-regularized regression to build the models for classifying AD patients for comparison. Table 7 shows that GenEpi is an efficient method,

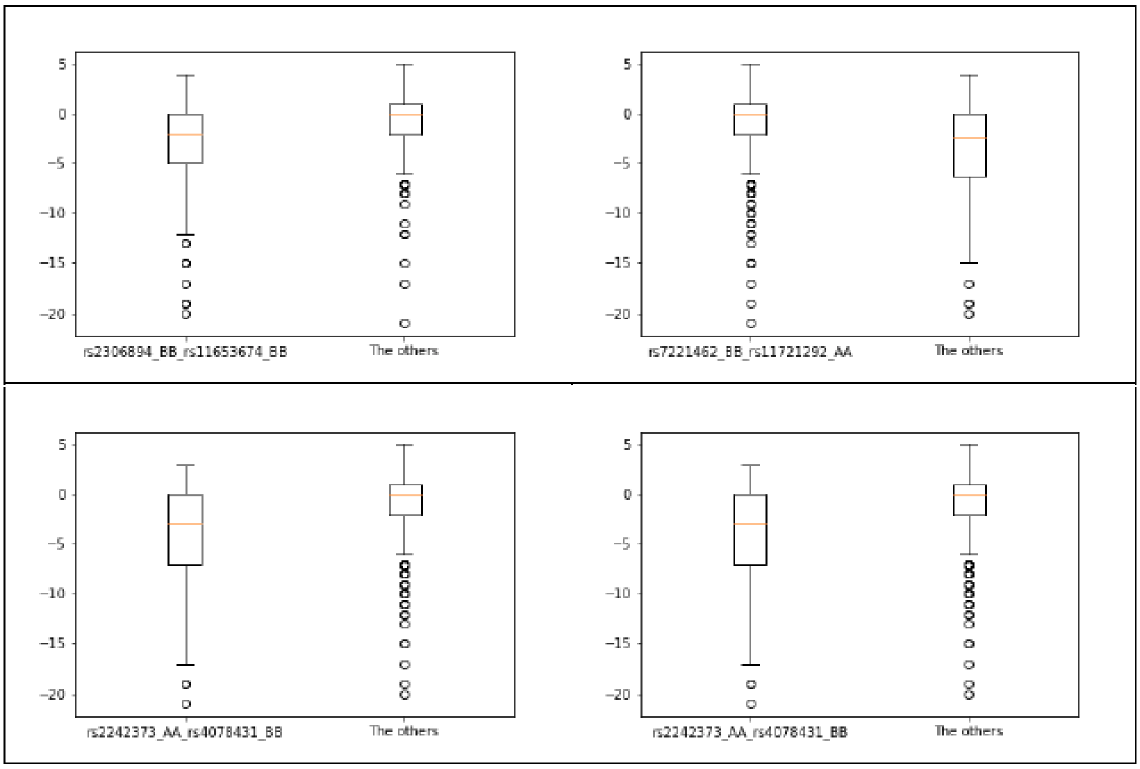

Figure 11. The box plots for combinations of SNPs from the genetic features selected by GenEpi in predicting the progression of AD. The y-axes of the plot are MMSE difference between ‘24-month follow-up MMSE’ and ‘MMSE at baseline’.

doi:10.6342/NTU202000613

Table 7. The comparison of different algorithms.

Algorithm	# Input SNP	Time Cost	Top 15	Top 30	Top 45	Top 60
GenEpi	4,916,249	9.95	0.76	0.72	0.71	0.68
BOOST	4,916,249	2157.6	0.31	0.24	0.30	0.37
ReliefF	33,868	0.11	0.52	0.48	0.45	0.46
FastEpistasis	12,809,667	836.8	0.62	0.61	0.60	0.59

'Time Cost' is the time spent on identifying the epistasis, which was measured by single CPU time in days.

The values in column top 15, 30, 45 and 60 are the 2-fold CV scores, which are the F1 scores.

which can deliver satisfied results for the epistasis discovery of 4 millions of SNPs within 9.95 CPU-days. The comparison of execution time is unfair to FastEpistasis, since FastEpistasis used the whole set of SNPs, which is about 2.6 time larger than the subset of it be used in GenEpi. When accuracy is considered, GenEpi has the best prediction performance despite the fact that GenEpi only uses the subset of SNPs from the final model. GenEpi shows that the time needed for identifying epistasis can be drastically

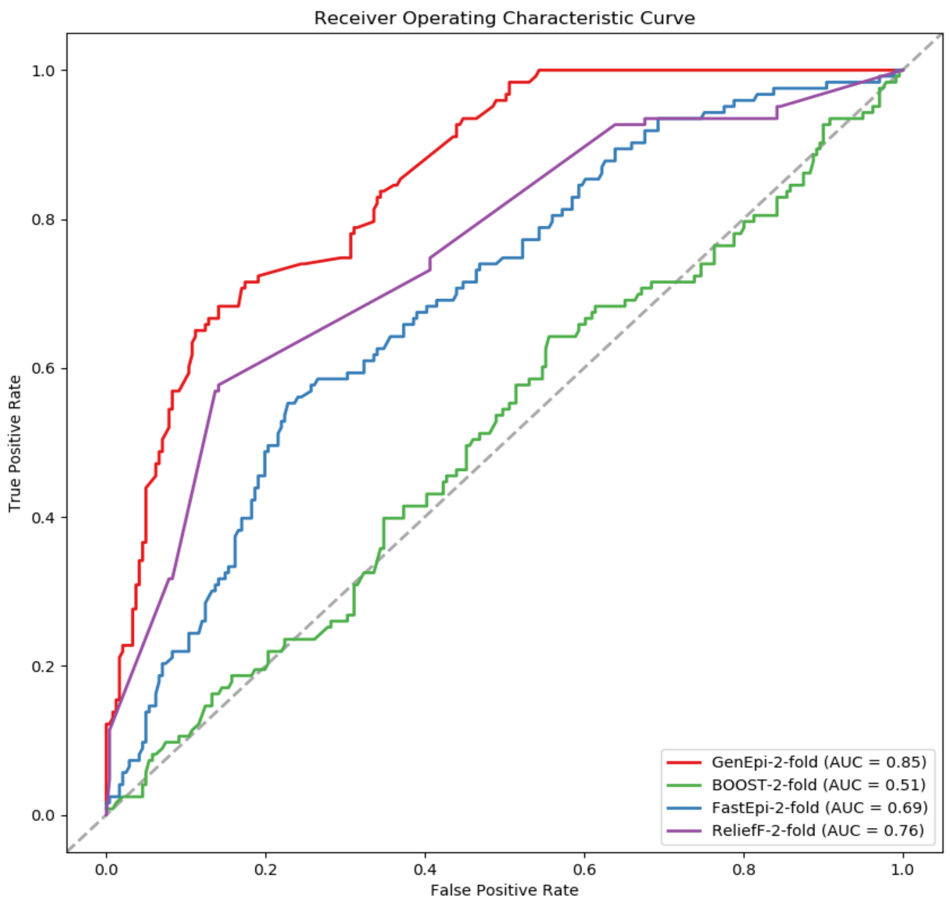

Figure 12. The ROC curves of different algorithms.

doi:10.6342/NTU202000613

reduced, without compromise to the performance. We provided the ROC curves for the classification task in Figure 12, and it shows that GenEpi achieved the best performance in double 2-fold CV procedures, of which the area under the curve (AUC) is 0.85.

doi:10.6342/NTU202000613

5. Discussion

The results in the previous section revealed the power of GenEpi to identify phenotype-associated epistasis efficiently. GenEpi revealed 14 features in 12 gens for predicting control or patient with AD and 4 features in 6 genes for predicting the progression of AD. Since AD is a chronic neurodegenerative disease, our findings would be supported if the gene identified by GenEpi are expressed in brains. We downloaded the median RPKM by tissue dataset (GTEx Analysis V6: dbGaP Accession phs000424.v6.p1) of the GTEx Project [99] and plotted a heatmap to inspect the gene expression of these genes in different tissues, as shown in Figure 13. Among the 18 genes selected by GenEpi, 13 have high expression level in the brain tissues. In addition, five genes, *VSNL1*, *PRKAG2-AS1*, *SYT6*, *CLEC1A* and *LINC00299*, have a similar expression pattern with *APOE*. For further investigation of the functions of these genes, we categorize them as cross-gene epistasis, single-gene epistasis and single SNP features base on the definition of GenEpi.

5.1. The cross-gene epistasis of AD selected by GenEpi

GenEpi picked out five cross gene epistasis in total, which include *CLEC1A * GPR142*, *CLEC1A * PRKAG2-AS1*, *IFT20 * RPL32*, *MFSD6L * PRKAG2-AS1* and *MICB * TOB2*. Liddelow, *et al.*[100] found that neurotoxic reactive astrocytes are induced by activated microglia, which would interact with AD [101]. *CLEC1A*, a phagocytic receptor gene, increased expression of genes in aged microglia, indicative of a pro-inflammatory status [102]. Based on database of HaploReg [103], we found that *GPR142* is DNAse region in primary astrocyte. It showed *CLEC1A* and *GPR142* are expressed in two inflammatory cells strongly related in AD. Li, *et al.* [104] found that *PRKAG2-AS1* and *PRKAG2* were differentially regulated. The *PRKAG2*, AMP-activated protein kinase subunit gamma-2, had a polymorphism associated with cognitive impairment as well as diabetes in old age

doi:10.6342/NTU202000613

[105]. AMPK activation is also related to cognitive impairment in old age, neuroinflammation, and neurodegeneration [106], and is subjective to a negative regulation by *PRKAG2-AS1* that thus may functionally interact with *CLEC1A* through AMPK regulated neuroinflammation. We did not find direct link between *RPL32* and AD, but by the data presented in HaploReg there is another gene called *IQSEC1* that located in same LD block. *IQSEC1* is expressed in neuron and physically binds to *APOE*, but the implication of this interaction in AD is not clear. *IFT20* is an essential component of primary cilia that appears to be important in the survival of dentate granule cell, which plays a key role in memory formation and related to AD [107]. These studies showed that

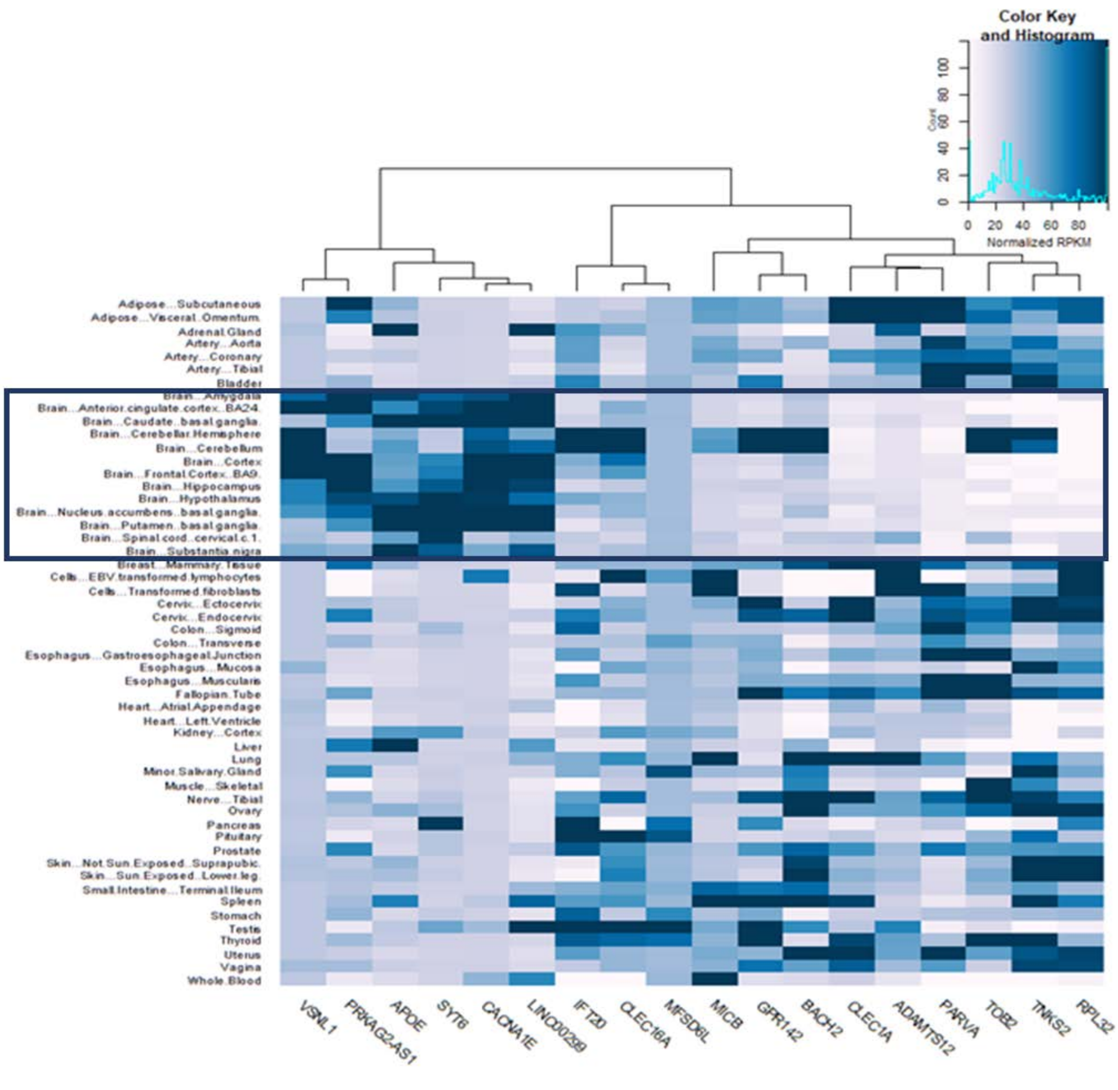

Figure 13. The heatmap of gene expression in different tissues for the 18 genes selected by GenEpi. The blue box highlights the sub-regions of brain.

doi:10.6342/NTU202000613

IFT20 and *IQSEC1* have been independently implicated in AD, how they functionally interact together in AD is not clear.

GenEpi picked out only one cross-gene epistasis for predicting control or patient with AD, which is MICB * TOB2. MIC proteins are induced on the cell surface by stress and are ligands of a common activator natural killer-cell receptor (*NKG2D*) [108]. They thus have roles in antimicrobial defense, tumor suppression and autoimmune-like diseases [109]. Quiroga et al. has found that the human MHC class I chain-related genes (*MICA* and *MICB*) are located within the HLA region that has stratified association with AD in the cohort of the Oxford Project to Investigate Memory and Ageing (OPTIMA) even though the association did not survive the correction for multiple testing [110]. *TOB2* has been reported as a most up-regulated gene [111] by expression analysis on multiple AD expression datasets from the GEO database to identify significant genes associated with electrophysiological pathways and attempted determination of interconnected canonical molecular pathways. The interaction of *MICB* and *TOB2* suggests a not-yet-explored biological process in AD pathogenesis. However, the significant associations of these genes in genome-wide association studies associated with immune system diseases and neurological diseases (see GWAS Catalog, 22561518, 29083406, 28540026), warrant future investigation. For all of the cross epistasis which selected by GenEpi, only one gene *MFSD6L* cannot find the association with AD so far.

5.2. The single-gene epistasis of AD selected by GenEpi

There are several possible modes of action accounting for intramolecular SNP-SNP interactions identified in this study. First: a synergistic regulation of transcription. For example, rs12095538 resides in the eQTL of STY6 and rs2774308 resides in a DNAse region of STY6. It is very possible that variants in both rs12095538 and rs2774308

doi:10.6342/NTU202000613

synergistically affect the expression of STY6. SYT6 encodes protein control neurotransmitter release. It has been shown that the expression of SYT6 is significantly down regulated in AD [112], supporting the causative role of the inherited rs12095538 and rs2774308 haplotype in AD progression. Likewise, rs56233035 and rs3678 reside in the LD blocks containing the eQTL and DNAse region of CACNA1E, respectively. CACNA1E encodes a subunit of CaV1.2 calcium channel. Its expression level in reactive astrocytes is associated with amyloid-β plaques formation in an AD mouse model [113]. Other examples are the interactions between rs12926153 and rs12922908, and rs9652600 and rs12922908 that are within DNAse regions controlling the expression of CLEC16A in brain. CLEC16A encodes a lectin receptor involved in inflammation process and is differentially expressed in AD brains [114]. Lastly, rs2052573 and rs34580133 reside in DNAse regions within the LINC00299 and thus may synergistically control the expression of this long non-coding RNA that has been implicated in brain development [115].

Second: a synergestic interaction between transcriptional and post-transcriptional regulation. For example, rs9344977, in an intron of BACH2 containing cis-regulatory elements, may affect the expression of BACH2, whereas rs56148686 in another intron of BACH2 without any cis-regulatory element. We suspect that rs56148686 may interact with s9344977 through a post-transcriptional mechanism such as RNA stability, RNA splicing, or microRNA binding. BACH2 encodes a transcription factor involving cellular responses toward oxidative stress. Interestingly, BACH2 is upregulated in cultured human neuroblastoma cells upon exposure to Alzheimer Amyloid β [116]. Similar mechanisms may apply to interaction pairs rs12189429-rs6881360 and rs12187423-rs6881360 that presumably regulating the abundance of ADAMTS12 mRNA. The importance of ADAMTS12 in AD is supported by another independent study using same

doi:10.6342/NTU202000613

ANDI dataset [117]. Third: an intramolecular SNP pairs modulating the expression of two separate neighboring genes. For example, rs11675339 and rs2710687 of VSNL1 appear to control the expression of GEN1 and SMC6 respectively. We also found that VSNL (also known as VILIP-1) has strong evidences to AD [118-120]. GEN1 and SMC6 play pivotal roles in repairing double-strand breaks of genomic DNA. Interestingly, rDNA instability, a result of failures to repair double-strand breaks of genomic DNA, has been implicated in AD [121]. Most of the single-gene epistasis selected by GenEpi can be explained by these three possible modes and only two of the significance SNP-SNP interactions are not immediately clear at this moment. For example, rs12366151 resides in the eQTL of MICALCL, an AD associated gene [122] but the interacting rs10831829 resides in the intron of PARVA, a gene not-yet-reported to be associated with AD. Similarly, rs2421701 and rs200512701 reside in two eQTLs regulating TNKS2 expression. However, the ADP-ribose polymerase functionally contributes to AD is not clear. Moreover, there are only two single SNP feature and both of them are located in APOE, which is well-known causal gene of AD. Moreover, GeneEpi successfully selected out the SNP rs429358, which determine the allele type of APOE.

doi:10.6342/NTU202000613

6. Conclusions

This study presents GenEpi, a computational package, to uncover epistasis associated with phenotypes by the proposed machine learning approach, which adopts two-element combinatorial encoding when producing features and constructs the prediction models by L1-regularized regression with stability selection. And, the study made specific contributions to discovery epistasis. This study released an open-source python package GenEpi on The Python Package Index (PyPi) with clearly written documents. No matter on the simulation data or real data (ADNI), GenEpi consistently outperforms FastEpistasis, BOOST, and ReliefF. The results on the simulation data and the real data of AD demonstrated that GenEpi has the ability to detect the epistasis associated with phenotypes effectively and efficiently. Moreover, when applying GenEpi on the ADNI dataset to discover the AD-related multifactorial pathogenesis, the result suggested the additional genetic feature selected by GenEpi can improve the prediction performance. Convincingly, the selected genetic features included the SNP, rs429358, of the well-known disease marker APOE. Among the 18 genes selected by GenEpi, 13 have high expression level in the brain tissues. With the comprehensive literature survey conducted in this study, the references support that the selected genes are highly correlated to AD.

While GenEpi has shown its ability to identify epistasis efficiently, it might still have the following limitations. Firstly, GenEpi can only detect pairwise interactions. Considering the false positive rate and computational complexity, it may not be appropriate for continuously generating the high-dimension interactions. A feature engineering-free method such as deep learning could be applied for discovering the high-dimension interactions. Second, GenEpi is a memory-consuming package, which might cause memory errors when calculating the epistasis of a gene containing a large number of SNPs. We recommend that the memory for running GenEpi should be over 256 GB.

doi:10.6342/NTU202000613

Since most features may not be associated with the phenotype, additional filters for feature selection can be designed to further reduce the number of features before modeling. Finally, a small sample size may lead overfitting, which forces us to use strict thresholds during feature selection. In this way, GenEpi delivers a high precision rate, but might suffer from false negatives. This implies different GWAS data might detect different sets of true positives. In traditional GWAS, meta-analysis [123] can be used to identify the common effects from multiple studies. This post statistical procedure could be considered for obtaining a common set from multiple GWAS data.

In summary, the results of this study demonstrated that GenEpi is a promising software package to identify phenotype-related SNPs and epistasis in GWAS, and it can be further used to predict the phenotypes. The release package GenEpi is an open-source Python package and available free of charge for non-commercial users. The package has been published on PyPi, and GitHub (https://github.com/Chester75321/GenEpi), and can be generalized to largely facilitate the studies of many complex diseases in the near future.

doi:10.6342/NTU202000613

References

1. Kingsmore, S.F., et al., *Genome-wide association studies: progress and potential for drug discovery and development.* Nature Reviews Drug Discovery, 2008. **7**(3): p. 221-230.
2. Ozaki, K., et al., *Functional SNPs in the lymphotoxin-alpha gene that are associated with susceptibility to myocardial infarction.* Nat Genet, 2002. **32**(4): p. 650-4.
3. Klein, R.J., et al., *Complement factor H polymorphism in age-related macular degeneration.* Science, 2005. **308**(5720): p. 385-9.
4. Pinero, J., et al., *DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants.* Nucleic Acids Res, 2017. **45**(D1): p. D833-D839.
5. McCarthy, M.I., et al., *Genome-wide association studies for complex traits: consensus, uncertainty and challenges.* Nat Rev Genet, 2008. **9**(5): p. 356-69.
6. Eichler, E.E., et al., *Missing heritability and strategies for finding the underlying causes of complex disease.* Nat Rev Genet, 2010. **11**(6): p. 446-50.
7. Manolio, T.A., et al., *Finding the missing heritability of complex diseases.* Nature, 2009. **461**(7265): p. 747-53.
8. Frazer, K.A., et al., *Human genetic variation and its contribution to complex traits.* Nat Rev Genet, 2009. **10**(4): p. 241-51.
9. Shriner, D., et al., *Problems with genome-wide association studies.* Science, 2007. **316**(5833): p. 1840-2.
10. Carlborg, O. and C.S. Haley, *Epistasis: too often neglected in complex trait studies?* Nat Rev Genet, 2004. **5**(8): p. 618-25.
11. Cordell, H.J., *Detecting gene-gene interactions that underlie human diseases.* Nat Rev Genet, 2009. **10**(6): p. 392-404.
12. Easton, D.F., et al., *Genome-wide association study identifies novel breast cancer susceptibility loci.* Nature, 2007. **447**(7148): p. 1087-93.

doi:10.6342/NTU202000613

13. Bodmer, W. and C. Bonilla, *Common and rare variants in multifactorial susceptibility to common diseases.* Nat Genet, 2008. **40**(6): p. 695-701.

14. Hindorff, L.A., et al., *Potential etiologic and functional implications of genome-wide association loci for human diseases and traits.* Proc Natl Acad Sci U S A, 2009. **106**(23): p. 9362-7.

15. Moore, J.H., F.W. Asselbergs, and S.M. Williams, *Bioinformatics challenges for genome-wide association studies.* Bioinformatics, 2010. **26**(4): p. 445-55.

16. Marchini, J., P. Donnelly, and L.R. Cardon, *Genome-wide strategies for detecting multiple loci that influence complex diseases.* Nat Genet, 2005. **37**(4): p. 413-7.

17. Wei, W.H., G. Hemani, and C.S. Haley, *Detecting epistasis in human complex traits.* Nat Rev Genet, 2014. **15**(11): p. 722-33.

18. Schupbach, T., et al., *FastEpistasis: a high performance computing solution for quantitative trait epistasis.* Bioinformatics, 2010. **26**(11): p. 1468-9.

19. Wan, X., et al., *BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies.* Am J Hum Genet, 2010. **87**(3): p. 325-40.

20. Purcell, S., et al., *PLINK: a tool set for whole-genome association and population-based linkage analyses.* Am J Hum Genet, 2007. **81**(3): p. 559-75.

21. Chang, C.C., et al., *Second-generation PLINK: rising to the challenge of larger and richer datasets.* Gigascience, 2015. **4**: p. 7.

22. Moore, J.H. and S.M. Williams, *New strategies for identifying gene-gene interactions in hypertension.* Ann Med, 2002. **34**(2): p. 88-95.

23. Yang, P., et al., *Gene-gene interaction filtering with ensemble of filters.* BMC Bioinformatics, 2011. **12 Suppl 1**: p. S10.

24. Bureau, A., et al., *Identifying SNPs predictive of phenotype using random forests.* Genet Epidemiol, 2005. **28**(2): p. 171-82.

25. Schwarz, D.F., I.R. Konig, and A. Ziegler, *On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data.* Bioinformatics, 2010. **26**(14): p. 1752-8.

26. Wan, X., et al., *MegaSNPHunter: a learning approach to detect disease*

doi:10.6342/NTU202000613

predisposition SNPs and high level interactions in genome wide association study. BMC Bioinformatics, 2009. **10**: p. 13.

27. Murk, W. and A.T. DeWan, *Exhaustive Genome-Wide Search for SNP-SNP Interactions Across 10 Human Diseases.* G3 (Bethesda), 2016. **6**(7): p. 2043-50.
28. Allen, G.I., et al., *Crowdsourced estimation of cognitive decline and resilience in Alzheimer's disease.* Alzheimers Dement, 2016. **12**(6): p. 645-53.
29. Bettens, K., K. Sleegers, and C. Van Broeckhoven, *Genetic insights in Alzheimer's disease.* The Lancet Neurology, 2013. **12**(1): p. 92-104.
30. Levy, E., et al., *Mutation of the Alzheimer's disease amyloid gene in hereditary cerebral hemorrhage, Dutch type.* Science, 1990. **248**(4959): p. 1124-1126.
31. Goate, A., et al., *Segregation of a missense mutation in the amyloid precursor protein gene with familial Alzheimer's disease.* Nature, 1991. **349**(6311): p. 704.
32. Levy-Lahad, E., et al., *Candidate gene for the chromosome 1 familial Alzheimer's disease locus.* Science, 1995. **269**(5226): p. 973-977.
33. Rogaev, E., et al., *Familial Alzheimer's disease in kindreds with missense mutations in a gene on chromosome 1 related to the Alzheimer's disease type 3 gene.* Nature, 1995. **376**(6543): p. 775.
34. Denise Harold, e.a., *Genome-wide association study identifies variants at CLU and PICALM associated with Alzheimer's disease.* Nature Genetics, 2009. **41.10**: p. 1088-1093.
35. Teri A. Manolio, M.D., Ph.D, *Genomewide Association Studies and Assessment of the Risk of Disease.* N Engl J Med, 2010. **363**: p. 166-176.
36. Sudha Seshadri, M.A.L.F., PhD; M. Arfan Ikram, MD, PhD; et al, *Genome-wide Analysis of Genetic Loci Associated With Alzheimer Disease.* JAMA, 2010. **303**: p. 1832-1840.
37. Hollingworth P, e.a., *Common variants at ABCA7, MS4A6A/MS4A4E, EPHA1, CD33 and CD2AP are associated with Alzheimer's disease.* Nature Genetics, 2011. **43**: p. 429-435.
38. Adam C Naj, e.a., *Common variants at MS4A4/MS4A6E, CD2AP, CD33 and*

doi:10.6342/NTU202000613

EPHA1 are associated with late-onset Alzheimer's disease. Nature Genetics, 2011. **43**: p. 436-441.

39. Jean-Charles Lambert, e.a., *Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer's disease.* Nature Genetics, 2013. **45**: p. 1452-1458.

40. Genevera I. Allen, e.a., *Crowdsourced estimation of cognitive decline and resilience in Alzheimer's disease.* Alzheimer's & Dementia-, 2016. **12**(6): p. 645-653.

41. Jennifer Couzin, e.a., *What don't we know?* Science, 2005. **309**: p. 78-102.

42. Stephen F. Kingsmore, I.E.L., Joann Mudge, Damian D. Gessler and William D. Beavis, *Genome-wide association studies: progress and potential for drug discovery and development.* Nature Review, 2008. **7**: p. 221-230.

43. Kouichi Ozaki, e.a., *Functional SNPs in the lymphotoxin-α gene that are associated with susceptibility to myocardial infarction.* Nature Genetics, 2002. **32**: p. 650-654.

44. Robert J. Klein, e.a., *Complement Factor H Polymorphism in Age-Related Macular Degeneration.* Science, 2005. **308**(5720): p. 385-389.

45. Janet Piñero, e.a., *DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants.* Nucleic Acids Res, 2107. **45**(D1): p. 833-839.

46. Mark I. McCarthy, e.a., *Genome-wide association studies for complex traits: consensus, uncertainty and challenges.* Nature Reviews Genetics, 2008. **9**: p. 356-369.

47. Douglas F. Easton, e.a., *Genome-wide association study identifies novel breast cancer susceptibility loci.* Nature, 2007. **447**: p. 1087-1093.

48. Daniel Shriner, e.a., *Problems with Genome-Wide Association Studies.* Science, 2007. **316**(5833): p. 1840-1842.

49. Sheila M. Schmutz, T.G.B., Angela D. Goldfinch, *TYRP1 and MC1R genotypes and their effects on coat color in dogs.* Mammalian Genome, 2002. **13**(7): p. 380-

doi:10.6342/NTU202000613

387.

50. J. A. Kerns, M.O., et al., *Exclusion of Melanocortin-1Receptor (Mc1r) and Agouti as Candidates for Dominant Black in Dogs.* Journal of Heredity, 2003. **94**(I): p. 75-79.

51. Bateson William, G.M., *Mendel's principles of heredity*. 1913: University press.

52. Phillips, P.C., *Epistasis — the essential role of gene interactions in the structure and evolution of genetic systems.* Nature Reviews Genetics, 2008. **9**: p. 855-867.

53. Cordell, H.J., *Detecting gene–gene interactions that underlie human diseases.* Nature Reviews Genetics, 2009. **10**: p. 392-404.

54. Walter Bodmer, C.B., *Common and rare variants in multifactorial susceptibility to common diseases.* Nature Genetics, 2008. **40**: p. 695-701.

55. Kelly A. Frazer, S.S.M., Nicholas J. Schork, Eric J. Topol, *Human genetic variation and its contribution to complex traits.* Nature Reviews Genetics, 2009. **10**: p. 241-251.

56. Teri A. Manolio, e.a., *Finding the missing heritability of complex diseases.* Nature, 2009. **461**: p. 747-753.

57. Lucia A. Hindorff, e.a., *Potential etiologic and functional implications of genome-wide association loci for human diseases and traits.* PNAS, 2009. **106**: p. 9362-9367.

58. Jason H. Moore, F.W.A., Scott M. Williams, *Bioinformatics challenges for genome-wide association studies.* Bioinformatics, 2010. **26**(4): p. 445-455.

59. Jonathan Marchini, P.D., Lon R Cardon, *Genome-wide strategies for detecting multiple loci that influence complex diseases.* Nature Genetics, 2005. **37**: p. 413-417.

60. Cordell, H.J., *Detecting gene–gene interactions that underlie human diseases.* Nature Reviews Genetics, 2009. **10**(6): p. 392-404.

61. Wen-Hua Wei, G.H., Chris S. Haley, *Detecting epistasis in human complex traits.* Nature Reviews Genetics, 2014. **15**: p. 722-733.

62. Shaun Purcell, e.a., *PLINK: A Tool Set for Whole-Genome Association and*

doi:10.6342/NTU202000613

Population-Based Linkage Analyses. The American Journal of Human Genetics, 2007. **81**: p. 559-575.

63. Christopher C Chang, e.a., *Second-generation PLINK: rising to the challenge of larger and richer datasets.* GigaScience, 2015. **4**.

64. Thierry Schüpbach, e.a., *FastEpistasis: a high performance computing solution for quantitative trait epistasis.* Bioinformatics, 2010. **26**: p. 1468-1469.

65. Xiang Wan, e.a., *BOOST: A Fast Approach to Detecting Gene-Gene Interactions in Genome-wide Case-Control Studies.* The American Society of Human Genetics, 2010. **87**: p. 325-340.

66. DeWan, W.M.a.A.T., *Exhaustive Genome-Wide Search for SNP-SNP Interactions Across 10 Human Diseases.* G3: Genes, Genomes, Genetics, 2016. **6**: p. 2043-2050.

67. Jason H Moore, S.M.W., *New strategies for identifying gene-gene interactions in hypertension.* Ann Med, 2002. **34**: p. 88-95.

68. Tricia A. Thornton-Wells, J.H.M.a.J.L.H., *Genetics, statistics and human disease: analytical retooling for complexity.* TRENDS in Genetics, 2004. **20**: p. 640-647.

69. Wahlsten, D., *Insensitivity of the analysis of variance to heredity-environment interaction.* Behavioral and Brain Sciences, 1990. **13**(1): p. 109-120.

70. Rosanna Upstill-Goddard, D.E., Joerg Fliege and Andrew Collins, *Machine learning approaches for the discovery of gene^gene interactions in disease data.* Briefings in Bioinformatics, 2012. **14**: p. 251-260.

71. Ho, T.K., *Random Decision Forests*, in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. 1995: Montreal, QC. p. 278-282.

72. Ho, T.K., *The Random Subspace Method for Constructing Decision Forests*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998. p. 832-844.

73. Pengyi Yang, J.W.H., Yee Hwa Yang, Bing B Zhou, *Gene-gene interaction filtering with ensemble of filters.* BMC Bioinformatics, 2011. **12:(Suppl 1):S10**.

doi:10.6342/NTU202000613

74. Bureau, A., *Identifying SNPs predictive of phenotype using random forests.* Genetic Epidemiology, 2005. **28**: p. 171-182.

75. Daniel F. Schwarz, I.R.K.a.A.Z., *On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data.* Bioinformatics, 2010. **26**: p. 1752-1758.

76. David M. Reif, e.a., *Feature Selection using a Random Forests Classifier for the Integrated Analysis of Multiple Data Types*, in *Computational Intelligence and Bioinformatics and Computational Biology*. 2006, IEEE: Toronto, Ont., Canada.

77. L Breiman, J.F., CJ Stone, RA Olshen, *Classification and regression trees*. 1984, Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. 358.

78. Mingers, J., *An empirical comparison of selection measures for decision-tree induction.* Machine Learning, 1989. **3**(4): p. 319-342.

79. Quinlan, J.R., *Discovering rules by induction from large collections of examples*. 1979: Expert systems in the micro electronic age. Edinburgh University Press. 168-201.

80. Kathryn L Lunetta, e.a., *Screening large-scale association study data: exploiting interactions using random forests.* BMC Genetics, 2004. **5:32**.

81. Xiang Wan, e.a., *MegaSNPHunter: a learning approach to detect disease predisposition SNPs and high level interactions in genome wide association study.* BMC Bioinformatics, 2009. **10:13**.

82. Xiang-Yang Lou, e.a., *A Generalized Combinatorial Approach for Detecting Gene-by-Gene and Gene-by-Environment Interactions with Application to Nicotine Dependence.* Am. J. Hum. Genet, 2007. **80**: p. 1125-1137.

83. al., D.R.V.e., *A Balanced Accuracy Function for Epistasis Modeling in Imbalanced Datasets using Multifactor Dimensionality Reduction.* Genetic Epidemiology, 2007. **31**: p. 306-315.

84. Kristine A. Pattin, e.a., *A Computationally Efficient Hypothesis Testing Method for Epistasis Analysis Using Multifactor Dimensionality Reduction.* Genetic Epidemiology, 2009. **33**: p. 87-94.

doi:10.6342/NTU202000613

85. Kira, K.a.R., L.A., *A practical approach to feature selection*, in *Machine Learning: Proceedings of the AAAI'92*. 1992: San Francisco.

86. Ma, L., A.G. Clark, and A. Keinan, *Gene-based testing of interactions in association studies of quantitative traits.* PLoS Genet, 2013. **9**(2): p. e1003321.

87. Oh, S., et al., *A novel method to identify high order gene-gene interactions in genome-wide association studies: gene-based MDR.* BMC Bioinformatics, 2012. **13 Suppl 9**: p. S5.

88. Li, S. and Y. Cui, *Gene-centric gene–gene interaction: A model-based kernel machine method.* The Annals of Applied Statistics, 2012. **6**(3): p. 1134-1161.

89. Wu, X., et al., *A novel statistic for genome-wide interaction analysis.* PLoS Genet, 2010. **6**(9): p. e1001131.

90. Kent, W.J., et al., *The human genome browser at UCSC.* Genome Res, 2002. **12**(6): p. 996-1006.

91. Rosenbloom, K.R., et al., *The UCSC Genome Browser database: 2015 update.* Nucleic Acids Res, 2015. **43**(Database issue): p. D670-81.

92. Lewontin, R. and K.i. Kojima, *The evolutionary dynamics of complex polymorphisms.* Evolution, 1960. **14**(4): p. 458-472.

93. Lewontin, R.C., *The Interaction of Selection and Linkage. I. General Considerations; Heterotic Models.* Genetics, 1964. **49**(1): p. 49-67.

94. Hill, W.G., *Estimation of linkage disequilibrium in randomly mating populations.* Heredity, 1974. **33**(2): p. 229.

95. Friedman, J., T. Hastie, and R. Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent.* J Stat Softw, 2010. **33**(1): p. 1-22.

96. Meinshausen, N. and P. Bühlmann, *Stability selection.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2010. **72**(4): p. 417-473.

97. Urbanowicz, R.J., et al., *Relief-Based Feature Selection: Introduction and Review.* arXiv preprint arXiv:1711.08421, 2017.

98. Urbanowicz, R.J., et al., *GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures.* BioData Min, 2012. **5**(1): p. 16.

doi:10.6342/NTU202000613

99. Consortium, G.T., *The Genotype-Tissue Expression (GTEx) project.* Nat Genet, 2013. **45**(6): p. 580-5.

100. Liddelow, S.A., et al., *Neurotoxic reactive astrocytes are induced by activated microglia.* Nature, 2017. **541**(7638): p. 481-487.

101. von Bernhardi, R. and G. Ramirez, *Microglia-astrocyte interaction in Alzheimer's disease: friends or foes for the nervous system?* Biol Res, 2001. **34**(2): p. 123-8.

102. Raj, D., et al., *Increased White Matter Inflammation in Aging- and Alzheimer's Disease Brain.* Front Mol Neurosci, 2017. **10**: p. 206.

103. Ward, L.D. and M. Kellis, *HaploReg: a resource for exploring chromatin states, conservation, and regulatory motif alterations within sets of genetically linked variants.* Nucleic Acids Res, 2012. **40**(Database issue): p. D930-4.

104. Li, P., et al., *Epoxyeicosatrienoic acids enhance embryonic haematopoiesis and adult marrow engraftment.* Nature, 2015. **523**(7561): p. 468.

105. Kim, E., et al., *AMPK gamma2 subunit gene PRKAG2 polymorphism associated with cognitive impairment as well as diabetes in old age.* Psychoneuroendocrinology, 2012. **37**(3): p. 358-65.

106. Peixoto, C.A., et al., *AMPK activation: Role in the signaling pathways of neuroinflammation and neurodegeneration.* Exp Neurol, 2017. **298**(Pt A): p. 31-41.

107. Whitfield, J.F., et al., *The Possible Roles of the Dentate Granule Cell's Leptin and Other Ciliary Receptors in Alzheimer's Neuropathology.* Cells, 2015. **4**(3): p. 253-74.

108. Vivier, E., E. Tomasello, and P. Paul, *Lymphocyte activation via NKG2D: towards a new paradigm in immune recognition?* Curr Opin Immunol, 2002. **14**(3): p. 306-11.

109. Collins, R.W., *Human MHC class I chain related (MIC) genes: their biological function and relevance to disease and transplantation.* Eur J Immunogenet, 2004. **31**(3): p. 105-14.

110. Quiroga, I., et al., *Association study of MICA and MICB in Alzheimer's disease.*

doi:10.6342/NTU202000613

Tissue Antigens, 2009. **74**(3): p. 241-3.

111. Mirza, Z. and N. Rajeh, *Identification Of Electrophysiological Changes In Alzheimer's Disease: A Microarray Based Transcriptomics And Molecular Pathway Analysis Study.* CNS Neurol Disord Drug Targets, 2017.

112. Saura, C.A., A. Parra-Damas, and L. Enriquez-Barreto, *Gene expression parallels synaptic excitability and plasticity changes in Alzheimer's disease.* Front Cell Neurosci, 2015. **9**: p. 318.

113. Daschil, N., et al., *CaV1.2 calcium channel expression in reactive astrocytes is associated with the formation of amyloid-beta plaques in an Alzheimer's disease mouse model.* J Alzheimers Dis, 2013. **37**(2): p. 439-51.

114. Porcellini, E., et al., *Alzheimer's disease gene signature says: beware of brain viral infections.* Immun Ageing, 2010. **7**: p. 16.

115. Talkowski, M.E., et al., *Disruption of a large intergenic noncoding RNA in subjects with neurodevelopmental disabilities.* Am J Hum Genet, 2012. **91**(6): p. 1128-34.

116. Uhrig, M., et al., *New Alzheimer amyloid beta responsive genes identified in human neuroblastoma cells by hierarchical clustering.* PLoS One, 2009. **4**(8): p. e6779.

117. Wang, W., et al., *A Multi-Marker Genetic Association Test Based on the Rasch Model Applied to Alzheimer's Disease.* PLoS One, 2015. **10**(9): p. e0138223.

118. Babic Leko, M., et al., *Predictive Value of Cerebrospinal Fluid Visinin-Like Protein-1 Levels for Alzheimer's Disease Early Detection and Differential Diagnosis in Patients with Mild Cognitive Impairment.* J Alzheimers Dis, 2016. **50**(3): p. 765-78.

119. Kirkwood, C.M., et al., *Altered Levels of Visinin-Like Protein 1 Correspond to Regional Neuronal Loss in Alzheimer Disease and Frontotemporal Lobar Degeneration.* J Neuropathol Exp Neurol, 2016. **75**(2): p. 175-82.

120. Luo, X., et al., *CSF levels of the neuronal injury biomarker visinin-like protein-1 in Alzheimer's disease and dementia with Lewy bodies.* J Neurochem, 2013.

doi:10.6342/NTU202000613

127(5): p. 681-90.

121. Pietrzak, M., et al., *Epigenetic silencing of nucleolar rRNA genes in Alzheimer's disease.* PLoS One, 2011. **6**(7): p. e22585.

122. Rouillard, A.D., et al., *The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins.* Database (Oxford), 2016. **2016**.

123. Lambert, J.C., et al., *Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer's disease.* Nat Genet, 2013. **45**(12): p. 1452-8.

doi:10.6342/NTU202000613

GenEpi Documentation

Chester (Yu-Chuan Chang)

Jan 02, 2020

doi:10.6342/NTU202000613

Contents

doi:10.6342/NTU202000613

GenEpi[1] is a package to uncover epistasis associated with phenotypes by a machine learning approach, developed by Yu-Chuan Chang at c4Lab of National Taiwan University and AILabs.

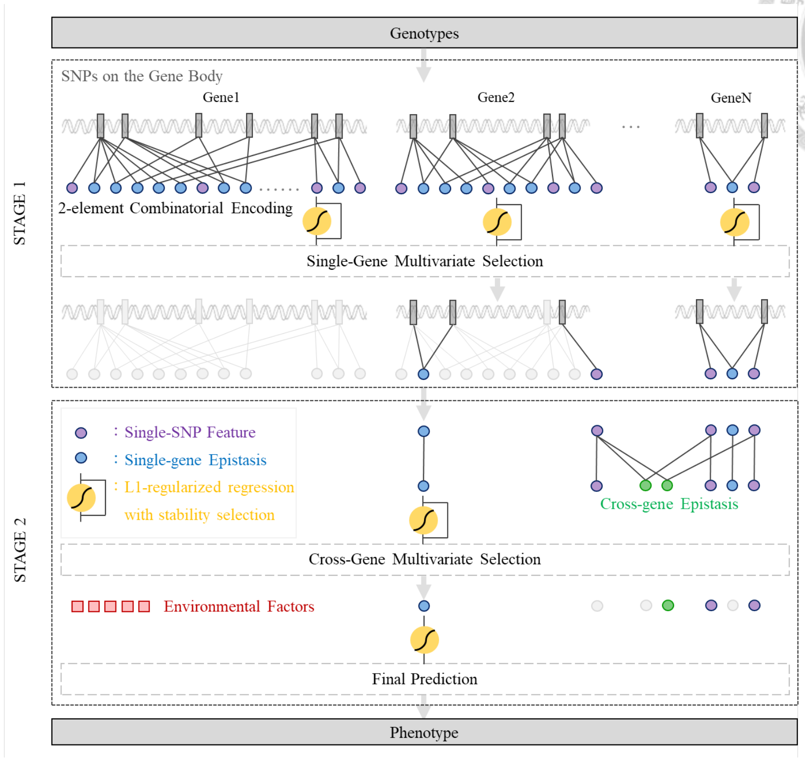

Fig. 1: The architecture and modules of GenEpi.

[1] Yu-Chuan Chang, June-Tai Wu, Ming-Yi Hong, Yi-An Tung, Ping-Han Hsieh, Sook Wah Yee, Kathleen M. Giacomini, Yen-Jen Oyang, and Chien-Yu Chen. "Genepi: Gene-Based Epistasis Discovery Using Machine Learning." bioRxiv (2019): 421719.

doi:10.6342/NTU202000613

CHAPTER 1

Introduction

GenEpi is designed to group SNPs by a set of loci in the gnome. For examples, a locus could be a gene. In other words, we use gene boundaries to group SNPs. A locus can be generalized to any particular regions in the genome, e.g. promoters, enhancers, etc. GenEpi first considers the genetic variants within a particular region as features in the first stage, because it is believed that SNPs within a functional region might have a higher chance to interact with each other and to influence molecular functions.

GenEpi adopts two-element combinatorial encoding when producing features and models them by L1-regularized regression with stability selection In the first stage (STAGE 1) of GenEpi, the genotype features from each single gene will be combinatorically encoded and modeled independently by L1-regularized regression with stability selection. In this way, we can estimate the prediction performance of each gene and detect within-gene epistasis with a low false positive rate. In the second stage (STAGE 2), both of the individual SNP and the within-gene epistasis features selected by STAGE 1 are pooled together to generate cross-gene epistasis features, and modeled again by L1-regularized regression with stability selection as STAGE 1. Finally, the user can combine the selected genetic features with environmental factors such as clinical features to build the final prediction models.

doi:10.6342/NTU202000613

CHAPTER 2

Citing

Please considering cite the following paper (currently avaliables as a pre-print in BioRxiv) if you use GenEpi in a scientific publication.

2.1 Installation

2.1.1 How to Install

GenEpi supports Python 3.7 and up. Use pip to install GenEpi and its dependencies.

```
$ pip install GenEpi
```

Check that you installed the GenEpi sucessfully.

```
$ GenEpi --help
```

After executed previous command on console, you will see:

```
usage: GenEpi [-h] -g G -p P [-s S] [-o O] [-m {c,r}] [-k K] [-t T]
              [--updatedb] [-b {hg19,hg38}] [--compressld] [-d D] [-r R]

optional arguments:
  -h, --help      show this help message and exit
  -g G            filename of the input .gen file
  -p P            filename of the input phenotype
  -s S            self-defined genome regions
  -o O            output file path
  -m {c,r}        choose model type: c for classification; r for regression
  -k K            k of k-fold cross validation
  -t T            number of threads

update UCSC database:
```

(continues on next page)

doi:10.6342/NTU202000613

(continued from previous page)

```
  --updatedb      enable this function
  -b {hg19,hg38}  human genome build

compress data by LD block:
  --compressld    enable this function
  -d D            threshold for compression: D prime
  -r R            threshold for compression: R square
```

2.1.2 Dependencies

Here is the dependency list for running GenEpi. pip takes care of these dependencies automatically when you install GenEpi.

- numpy >= 1.13.0
- psutil >= 4.3.0
- pymysql >= 0.8.0
- scipy >= 0.19.0
- scikit-learn>=0.21.2

2.1.3 System Requirements

For running a quick test, you could install GenEpi on any laptop e.g. a MacBook. When applying GenEpi on a real whole genome-wide dataset, here are recommended system requirements:

Processor	2.3 GHz Intel XEON® E5-2673 v4 * 32
RAM	256 GiB
Storage	500 GiB

These requirements are refer to the specification of Microsoft Azure E32 v3.

Note: GenEpi is a memory-consuming package, which might cause memory errors when calculating the epistasis of a gene containing a large number of SNPs. We recommend that the memory for running GenEpi should be over 256 GB.

2.2 Quickstart

This section gets you started quickly, the I/O described in I/O File Fomats, more usage examples please find in More Usage Examples, discussing each of the sub-modules introduced in How it Work.

2.2.1 Running a Quick Test

Please use following command to run a quick test, you will obtain all the outputs of GenEpi in your current folder.

```
$ GenEpi -g example -p example -o ./
```

The progress will print on console:

doi:10.6342/NTU202000613

```
step1: Down load UCSC Database. DONE!
step2: Estimate LD. DONE!
Warning of step3: .gen file should be sorted by chromosome and position
step3: Split by gene. DONE!
step4: Detect single gene epistasis. DONE!
step5: Detect cross gene epistasis. DONE! (Training score:0.63; 2-fold Test Score:0.
↪61)
step6: Ensemble with covariates. DONE! (Training score:0.63; 2-fold Test Score:0.60)
```

GenEpi will automatically generate three folders (snpSubsets, singleGeneResult, crossGeneResult) in output path (arg: -o). The following tree structure is the contents of the output folder. You could go to the folder **crossGeneResult** directly to obtain your main result table for episatasis in **Result.csv**.

```
./
├── GenEpi_Log_DATE-TIME.txt
├── crossGeneResult
│   ├── Classifier.pkl
│   ├── Classifier_Covariates.pkl
│   ├── Feature.csv
│   └── Result.csv
├── sample.LDBlock
├── sample.csv
├── sample.gen
├── sample_LDReduced.gen
├── singleGeneResult
│   ├── All_Logistic_k2.csv
│   ├── APOC1_Feature.csv
│   ├── APOC1_Result.csv
│   ├── APOE_Feature.csv
│   ├── APOE_Result.csv
│   ├── PVRL2_Feature.csv
│   ├── PVRL2_Result.csv
│   ├── TOMM40_Feature.csv
│   └── TOMM40_Result.csv
└── snpSubsets
    ├── APOC1_23.gen
    ├── APOE_11.gen
    ├── PVRL2_48.gen
    └── TOMM40_67.gen
```

2.2.2 Interpreting the Main Result Table

Here is the contents of Result.csv, which mean the episatasis seleted by GenEpi.

doi:10.6342/NTU202000613

RSID	Weight	-Log10(χ2 p-value)	Odds Ratio	Genotype Frequency	Gene Symbol
rs157580_BB rs2238681_AA	0.9729	8.4002	9.3952	0.1044	TOMM40
rs449647_AA rs769449_AB	0.7065	8.0278	5.0877	0.2692	APOE
rs59007384_BB rs11668327_AA	1.0807	8.0158	12.0408	0.0824	TOMM40
rs283811_BB rs7254892_AA	1.0807	8.0158	12.0408	0.0824	PVRL2
rs429358_AA	-0.7587	5.7628	0.1743	0.5962	APOE
rs73052335_AA rs429358_AA	-0.7289	5.6548	0.1867	0.5714	APOC1*APOE

We listed the statistical significance of the selected genetic features in Result.csv. The first column lists each feature by its RSID and the genotype (denoted as RSID_genotype), the pairwise epistatis features are represented using two SNPs. The weights in the second column were extracted from the L1-regularized regression model. The last column describes the genes where the SNPs are located according to the genomic coordinates. We used a star sign to denote the epistasis between genes. The p-values of the χ2 test (the quantitative task will use student t-test) are also included. The odds ratio significantly away from 1 also indicates whether the features are potential causal or protective genotypes. Since low genotype frequency may cause unreliable odds ratios, we also listed this information in the table.

2.3 I/O File Fomats

We provided test data sample.gen and sample.csv in example folder. After running a quick test, GenEpi will automatically copy these test data to output path and generate all of the output folders and files as following tree structure. Please see the following detail about the format of these input and output data.

```
./
├── GenEpi_Log_DATE-TIME.txt
├── crossGeneResult
│   ├── Classifier.pkl
│   ├── Classifier_Covariates.pkl
│   ├── Feature.csv
│   └── Result.csv
├── sample.LDBlock
├── sample.csv
├── sample.gen
├── sample_LDReduced.gen
├── singleGeneResult
│   ├── All_Logistic_k2.csv
│   ├── APOC1_Feature.csv
│   ├── APOC1_Result.csv
│   ├── APOE_Feature.csv
│   ├── APOE_Result.csv
│   ├── PVRL2_Feature.csv
│   ├── PVRL2_Result.csv
│   ├── TOMM40_Feature.csv
│   └── TOMM40_Result.csv
└── snpSubsets
    ├── APOC1_23.gen
    ├── APOE_11.gen
```

(continues on next page)

doi:10.6342/NTU202000613

(continued from previous page)

```
    ├── PVRL2_48.gen
    └── TOMM40_67.gen
```

2.3.1 Input: Genotype Data

Sample.gen is an example of genotype data. GenEpi takes the Genotype File Format (.GEN) used by Oxford statistical genetics tools, such as IMPUTE2 and SNPTEST as the input format for genotype data. If your files are in PLINK format (.BED/.BIM/.FAM) or 1000 Genomes Project text Variant Call Format (.VCF), you could use PLINK with the following command to convert them to the .GEN file.

If your files are in the .BED/.BIM/.FAM format.

```
$ plink --bfile prefixOfTheFilename --recode oxford --out prefixOfTheFilename
```

If your file is in the .VCF format.

```
$ plink --vcf filename.vcf --recode oxford --out prefixOfTheFilename
```

2.3.2 Input: Phenotype Data

Sample.gen is an example of phenotype data with environmental factor (e.g. the age of each sample). GenEpi takes the common .CSV file without header line as the input format for phenotype and environmental factor data. The last column of the file will be considered as the phenotype data (e.g. 1 or 0, which indicate case/control, respectively) and the other columns will be considered as the environmental factor data.

Warning: The sequential order of the phenotype data should be the same as that in the .GEN file.

2.3.3 Output: LDBlock File

GenEpi has a moudle, which can reduce the dimension of the input feature by estimating the linkage disequilibrium (LD). After performing dimension reduction, GenEpi will generate two files, a dimension-reduced .GEN file and a file containing LD blocks (.LDBlock file). Each row in the .LDBlock file indicates a LD block (see below for examples). The SNPs in front of colon signs are the representative SNPs of each LD block, and only these SNPs will be retained in the dimension-reduced .GEN file.

```
rs429358:rs429358
rs7412:rs7412
rs117656888:rs117656888
rs1081105:rs1081105
rs1081106:rs1081106,rs191315680
```

2.3.4 Output: SnpSubsets Folder

Since GenEpi is a gene-based epistasis discovering method, the input genotype will first be splited into group of each gene. The subsets of the .GEN file for each gene will be stored in the folder snpSubsets. The naming rule of the filename is GeneSymbol_NumberOfVariantsOnGene.gen

doi:10.6342/NTU202000613

2.3.5 Output: SingleGeneResult Folder

In first stage of GenEpi, all the .GEN file for each gene will be modeled gene by gene. Every models will output two kinds of data the GeneSymbol_Result.csv and GeneSymbol_Feature.csv. The format of GeneSymbol_Result.csv please refer to the section Interpreting the Main Result Table. The only difference is the GeneSymbol_Result.csv is a result table for single gene. Moreover, GeneSymbol_Feature.csv are the raw features corresponds to the episatsis in GeneSymbol_Result.csv. Beside these two types of files, there is a file named All_Logistic/Lasso.csv, which contains all the prediction scores of each gene, please see as following example.

```
$ head All_Logistic_k2.csv

GeneSymbol,F1Score
PVRL2,0.5745454545454547
APOC1,0.5681818181818181
TOMM40,0.602510460251046
APOE,0.592
```

2.3.6 Output: CrossGeneResult Folder

The results of the second stage of GenEpi - cross gene modeling will be generated in this folder. The formats are as same as the description in previous section *SingleGeneResult Folder*. Moreover, the final models will be persisted in this folder as Classifier/Regressor.pkl and Classifier/Regressor_Covariates.pkl, respectively. You could keep these models for future use without reconstructing them.

2.3.7 Output: Porcess Log

The performance of genetic feature only and genetic + environmental factor models will be logged into GenEpi_Log_DATE-TIME.txt. Other process information such as the setting of arguments, the time cost will also be recorded.

```
start analysis at: 20191009-17:54:45

Arguments in effect:
        -g (input genotype filename): ./sample.gen
        -p (input phenotype filename): ./sample.csv
        -s (self-defined genome regions): None
        -o (output filepath): ./

        -m (model type): Classification -k (k-fold cross validation): 2
        -t (number of threads): 4

        --updatedb (enable function of update UCSC database): True
        -b (human genome build): hg19

        --compressld (enable function of LD data compression): True
        -d (D prime threshold): 0.9
        -r (R square threshold): 0.9

Number of variants: 223
Number of samples: 364
Overall genetic feature performance (F1 score)
Training: 0.6307053941908715
Testing (2-fold CV): 0.6134453781512604
```

(continues on next page)

doi:10.6342/NTU202000613

(continued from previous page)

```
Ensemble with co-variate performance (F1 score)
Training: 0.632
Testing (2-fold CV): 0.6016260162601627

end analysis at: 20191009-17:54:58
```

2.3.8 Seld-defined Genome Regions

GenEpi supports seld-defined genome regions for first stage to subset the data. Please prepare your genome regions in .TXT with the columns [chromosome, start, end, strand, geneSymbol], for eample:

```
1,10873,14409,+,DDX11L1
1,14361,30370,-,WASH7P
1,34610,37081,-,FAM138F
1,68090,70008,+,OR4F5
...
```

2.4 More Usage Examples

For checking all the optional arguments, please use –help.

```
$ GenEpi --help
```

You will obtain the following argument list:

```
usage: GenEpi [-h] -g G -p P [-s S] [-o O] [-m {c,r}] [-k K] [-t T]
              [--updatedb] [-b {hg19,hg38}] [--compressld] [-d D] [-r R]

optional arguments:
  -h, --help      show this help message and exit
  -g G            filename of the input .gen file
  -p P            filename of the input phenotype
  -s S            self-defined genome regions
  -o O            output file path
  -m {c,r}        choose model type: c for classification; r for regression
  -k K            k of k-fold cross validation
  -t T            number of threads

update UCSC database:
  --updatedb      enable this function
  -b {hg19,hg38}  human genome build

compress data by LD block:
  --compressld    enable this function
  -d D            threshold for compression: D prime
  -r R            threshold for compression: R square
```

2.4.1 Applying on Your Data

```
$ GenEpi -g full_path_of_your_.GEN_file -p full_path_of_your_.CSV_file -o ./
```

doi:10.6342/NTU202000613

2.4.2 For Quantitative Study

GenEpi can support both case/control and quantitative studies, for quantitative studies please modify the parameter -m. You could download the test data and excute the following command.

```
$ GenEpi -g sample.gen -p sample_q.csv -o ./ -m r
```

2.4.3 Changing the Genome Build

For changing the build of USCS genome browser, please modify the parameter -b.

```
$ GenEpi -g example -p example -o ./ --updatedb -b hg38
```

2.4.4 Threshold for Dimension Reduction

You could modify the threshold for Linkage Disequilibrium dimension reduction by following command.

```
$ GenEpi -g example -p example -o ./ --compressld -d 0.9 -r 0.9
```

2.4.5 Self-defined Genome Region

Please prepare your self-defined genome region in this format. Then, use the parameter -s for applying it on your data.

```
$ GenEpi -s full_path_of_your_genome_region_file -g full_path_of_your_.GEN_file -p␣
↪full_path_of_your_.CSV_file -o ./
```

2.5 How it Works

The main procedures of GenEpi described in Introduction. In addition to the main procedures, two pre-processing steps are also implemented in GenEpi: retrieving the gene information from public databases and reducing the gene information from public databases and reducing the dimensionality of the features using linkage disequilibrium. All of the precedures be implemented as six steps in GenEpi submoudles. The interations bewteen the I/O and these submodules please find in the figure below. For API documentations of these submodules please find in next section.

2.6 API Documentations

2.6.1 GenEpi module

Created on Apr 2019

@author: Chester (Yu-Chuan Chang)

genepi.GenEpi.**ArgumentsParser**()

To obtain and parse the arguments from user.

Parameters **None** –

Returns argparse.ArgumentParser

doi:10.6342/NTU202000613

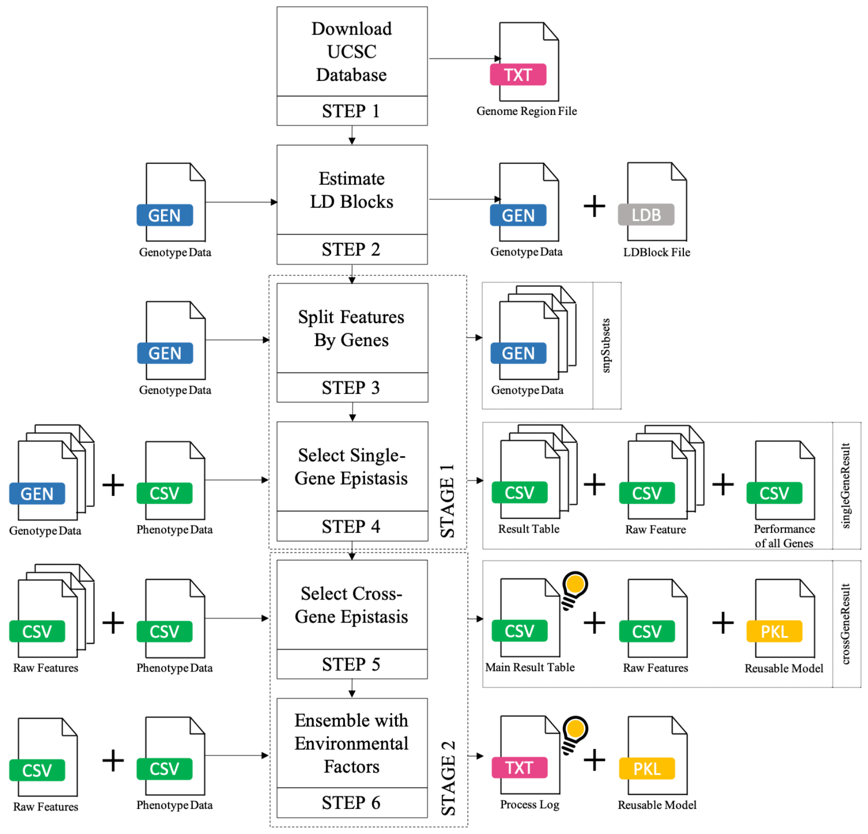

Fig. 1: The interations bewteen the I/O and GenEpi submodules.

doi:10.6342/NTU202000613

genepi.GenEpi.**InputChecking**(*str_inputFileName_genotype*, *str_inputFileName_phenotype*)
To check the numbers of sample are consistent in genotype and phenotype data.

Parameters

- **str_inputFileName_genotype** (*str*) – File name of input genotype data
- **str_inputFileName_phenotype** (*str*) – File name of input phenotype data

Returns

tuple containing:

- int_num_genotype (int): The sample number of genotype data
- int_num_phenotype (int): The sample number of phenotype data

Return type (tuple)

genepi.GenEpi.**main**(*args=None*)
Main function for obtaining user arguments, controling workflow and recording log file.

Parameters **None** –

Returns None

2.6.2 step1_downloadUCSCDB

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

genepi.step1_downloadUCSCDB.**DownloadUCSCDB**(*str_outputFilePath='/home/docs/checkouts/readthedocs.org/user_build*
packages/genepi-2.0.9-py3.7.egg/genepi',
str_hgbuild='hg19')
To obtain the gene information such as official gene symbols and genomic coordinates, this function is for retrieving kgXref and knownGene data table from the UCSC human genome annotation database

Parameters

- **str_outputFilePath** (*str*) – File path of output database
- **str_hgbuild** (*str*) – Genome build (eg. "hg19")

Returns

- Expected Success Response:

```
"step1: Down load UCSC Database. DONE!"
```

2.6.3 step2_estimateLD

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

genepi.step2_estimateLD.**EstimateAlleleFrequency**(*gen_snp*)
A function for estimating allele frequency of a single varaint

Parameters **gen_snp** (*list*) – The genotypes of a variant of all samples

Returns

tuple containing:

doi:10.6342/NTU202000613

- float_frequency_A (float): The reference allele type frequency
- float_frequency_B (float): The alternative allele type frequency

Return type (tuple)

`genepi.step2_estimateLD.`**`EstimateLDBlock`**(*str_inputFileName_genotype*, *str_outputFilePath=”*, *float_threshold_DPrime=0.8*, *float_threshold_RSquare=0.8*)

A function for implementing linkage disequilibrium (LD) dimension reduction. In genotype data, a variant often exhibits high dependency with its nearby variants because of LD. In the practical implantation, we prefer to group these dependent features to reduce the dimension of features. In other words, we can take the advantages of LD to reduce the dimensionality of genetic features. In this regard, this function adopted the same approach developed by Lewontin (1964) to estimate LD. We used D' and r2 as the criteria to group highly dependent genetic features as blocks. In each block, we chose the features with the largest minor allele frequency to represent other features in the same block.

Parameters

- **`str_inputFileName_genotype`** (*`str`*) – File name of input genotype data
- **`str_outputFilePath`** (*`str`*) – File path of output file
- **`float_threshold_DPrime`** (*`float`*) – The Dprime threshold for discriminating a LD block (default: 0.8)
- **`float_threshold_RSquare`** (*`float`*) – The RSquare threshold for discriminating a LD block (default: 0.8)

Returns

- Expected Success Response:

```
"step2: Estimate LD. DONE!"
```

`genepi.step2_estimateLD.`**`EstimatePairwiseLD`**(*gen_snp_1*, *gen_snp_2*)

Lewontin (1964) linkage disequilibrium (LD) estimation.

Parameters

- **`gen_snp_1`** (*`list`*) – The genotypes of first variant of all samples
- **`gen_snp_2`** (*`list`*) – The genotypes of second variant of all samples

Returns

tuple containing:

- float_D_prime (float): The DPrime of these two variants
- float_R_square (float): The RSquare of these two variants

Return type (tuple)

2.6.4 step3_splitByGene

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

doi:10.6342/NTU202000613

genepi.step3_splitByGene.**SplitByGene**(*str_inputFileName_genotype, str_inputFileName_UCSCDB='/home/docs/checkouts/readthedocs.org/user_bui packages/genepi-2.0.9-py3.7.egg/genepi/UCSCGenomeDatabase.txt', str_outputFilePath=''*)

In order to extract genetic features for a gene, this function used the start and end positions of each gene from the local UCSC database to split the genetic features. Then, generate the .GEN files for each gene in the folder named snpSubsets.

Parameters

- **str_inputFileName_genotype** (*str*) – File name of input genotype data
- **str_inputFileName_UCSCDB** (*str*) – File name of input genome regions
- **str_outputFilePath** (*str*) – File path of output file

Returns

- Expected Success Response:

```
"step3: Split by gene. DONE!"
```

Warning: "Warning of step3: .gen file should be sorted by chromosome and position"

genepi.step3_splitByGene.**SplitMegaGene**(*list_snpsOnGene, int_window, int_step, str_outputFilePath, str_outputFileName*)

In order to extract genetic features for a gene, this function used the start and end positions of each gene from the local UCSC database to split the genetic features. Then, generate the .GEN files for each gene in the folder named snpSubsets.

Parameters

- **list_snpsOnGene** (*list*) – A list contains SNPs on a gene
- **int_window** (*int*) – The size of the sliding window
- **int_step** (*int*) – The step of the sliding window
- **str_outputFilePath** (*str*) – File path of output file
- **str_outputFileName** (*str*) – File name of output file

Returns None

2.6.5 step4_singleGeneEpistasis_Lasso

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

genepi.step4_singleGeneEpistasis_Lasso.**BatchSingleGeneEpistasisLasso**(*str_inputFilePath_genotype, str_inputFileName_phenotype, str_outputFilePath='', int_kOfKFold=2, int_nJobs=4*)

Batch running for the single gene workflow.

Parameters

- **str_inputFilePath_genotype** (*str*) – File path of input genotype data

doi:10.6342/NTU202000613

- **str_inputFileName_phenotype** (*str*) – File name of input phenotype data
- **str_outputFilePath** (*str*) – File path of output file
- **int_kOfKFold** (*int*) – The k for k-fold cross validation (default: 2)
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns

- Expected Success Response:

```
"step4: Detect single gene epistasis. DONE!"
```

genepi.step4_singleGeneEpistasis_Lasso.**FeatureEncoderLasso**(*np_genotype_rsid*, *np_genotype*, *np_phenotype*, *int_dim*)

Implementation of the two-element combinatorial encoding.

Parameters

- **np_genotype_rsid** (*ndarray*) – 1D array containing rsid of genotype data with *str* type
- **np_genotype** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_phenotype** (*ndarray*) – 2D array containing phenotype data with *float* type
- **int_dim** (*int*) – The dimension of a variant (default: 3. AA, AB and BB)

Returns

tuple containing:

- list_interaction_rsid (ndarray): 1D array containing rsid of genotype data with *str* type
- np_interaction (ndarray): 2D array containing genotype data with *int8* type

Return type (tuple)

genepi.step4_singleGeneEpistasis_Lasso.**FilterInLoading**(*np_genotype*, *np_phenotype*)

This function is for filtering low quality varaint. Before modeling each subset of genotype features, two criteria were adopted to exclude low quality data. The first criterion is that the genotype frequency of a feature should exceed 5%, where the genotype frequency means the proportion of genotype among the total samples in the dataset. The second criterion is regarding the association between the feature and the phenotype. We used $\chi 2$ test to estimate the association between the feature and the phenotype, and the p-value should be smaller than 0.01.

Parameters

- **np_genotype** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_phenotype** (*ndarray*) – 2D array containing phenotype data with *float* type

Returns

np_genotype

2D array containing genotype data with *int8* type

Return type (ndarray)

doi:10.6342/NTU202000613

`genepi.step4_singleGeneEpistasis_Lasso.`**`LassoRegressionCV`**`(`*np_X*, *np_y*, *int_kOfKFold=2*, *int_nJobs=1*`)`

Implementation of the L1-regularized Lasso regression with k-fold cross validation.

Parameters

- **`np_X`** (*ndarray*) – 2D array containing genotype data with *int8* type
- **`np_y`** (*ndarray*) – 2D array containing phenotype data with *float* type
- **`int_kOfKFold`** (*int*) – The k for k-fold cross validation (default: 2)
- **`int_nJobs`** (*int*) – The number of thread (default: 1)

Returns

estimator.scores

1D array containing the scores of each genetic features with *float* type

Return type (ndarray)

`genepi.step4_singleGeneEpistasis_Lasso.`**`RandomizedLassoRegression`**`(`*np_X*, *np_y*`)`

Implementation of the stability selection.

Parameters

- **`np_X`** (*ndarray*) – 2D array containing genotype data with *int8* type
- **`np_y`** (*ndarray*) – 2D array containing phenotype data with *float* type

Returns

estimator.scores

1D array containing the scores of each genetic features with *float* type

Return type (ndarray)

`genepi.step4_singleGeneEpistasis_Lasso.`**`SingleGeneEpistasisLasso`**`(`*str_inputFileName_genotype*, *str_inputFileName_phenotype*, *str_outputFilePath=”*, *int_kOfKFold=2*, *int_nJobs=1*`)`

A workflow to model a single gene containing two-element combinatorial encoding, stability selection, filtering low quality varaint and L1-regularized Lasso regression with k-fold cross validation.

Parameters

- **`str_inputFileName_genotype`** (*str*) – File name of input genotype data
- **`str_inputFileName_phenotype`** (*str*) – File name of input phenotype data
- **`str_outputFilePath`** (*str*) – File path of output file
- **`int_kOfKFold`** (*int*) – The k for k-fold cross validation (default: 2)
- **`int_nJobs`** (*int*) – The number of thread (default: 1)

Returns

float_AVG_S_P

The average of the Peason’s and Spearman’s correlation of the model

Return type (float)

doi:10.6342/NTU202000613

2.6.6 step4_singleGeneEpistasis_Logistic

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

genepi.step4_singleGeneEpistasis_Logistic.**BatchSingleGeneEpistasisLogistic**(*str_inputFilePath_geno*
str_inputFileName_phe
str_outputFilePath=",
int_kOfKFold=2,
int_nJobs=4)

Batch running for the single gene workflow.

Parameters

- **str_inputFilePath_genotype** (*str*) – File path of input genotype data
- **str_inputFileName_phenotype** (*str*) – File name of input phenotype data
- **str_outputFilePath** (*str*) – File path of output file
- **int_kOfKFold** (*int*) – The k for k-fold cross validation (default: 2)
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns

- Expected Success Response:

```
"step4: Detect single gene epistasis. DONE!"
```

genepi.step4_singleGeneEpistasis_Logistic.**FeatureEncoderLogistic**(*np_genotype_rsid*,
np_genotype,
np_phenotype,
int_dim)

Implementation of the two-element combinatorial encoding.

Parameters

- **np_genotype_rsid** (*ndarray*) – 1D array containing rsid of genotype data with *str* type
- **np_genotype** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_phenotype** (*ndarray*) – 2D array containing phenotype data with *float* type
- **int_dim** (*int*) – The dimension of a variant (default: 3. AA, AB and BB)

Returns

tuple containing:

- list_interaction_rsid (ndarray): 1D array containing rsid of genotype data with *str* type
- np_interaction (ndarray): 2D array containing genotype data with *int8* type

Return type (tuple)

genepi.step4_singleGeneEpistasis_Logistic.**FilterInLoading**(*np_genotype*,
np_phenotype)

This function is for filtering low quality varaint. Before modeling each subset of genotype features, two criteria were adopted to exclude low quality data. The first criterion is that the genotype frequency of a feature should exceed 5%, where the genotype frequency means the proportion of genotype among the total samples in the dataset. The second criterion is regarding the association between the feature and the phenotype. We used $\chi 2$

doi:10.6342/NTU202000613

test to estimate the association between the feature and the phenotype, and the p-value should be smaller than 0.01.

Parameters

- **`np_genotype`** (*`ndarray`*) – 2D array containing genotype data with *int8* type
- **`np_phenotype`** (*`ndarray`*) – 2D array containing phenotype data with *float* type

Returns

np_genotype

2D array containing genotype data with *int8* type

Return type (ndarray)

`genepi.step4_singleGeneEpistasis_Logistic.`**`GenerateContingencyTable`**`(`*np_genotype*, *np_phenotype*`)`

Generating the contingency table for chi-square test.

Parameters

- **`np_X`** (*`ndarray`*) – 2D array containing genotype data with *int8* type
- **`np_y`** (*`ndarray`*) – 2D array containing phenotype data with *float* type

Returns

np_contingency

2D array containing the contingency table with *int* type

Return type (ndarray)

`genepi.step4_singleGeneEpistasis_Logistic.`**`LogisticRegressionL1CV`**`(`*np_X*, *np_y*, *int_kOfKFold=2*, *int_nJobs=1*`)`

Implementation of the L1-regularized Logistic regression with k-fold cross validation.

Parameters

- **`np_X`** (*`ndarray`*) – 2D array containing genotype data with *int8* type
- **`np_y`** (*`ndarray`*) – 2D array containing phenotype data with *float* type
- **`int_kOfKFold`** (*`int`*) – The k for k-fold cross validation (default: 2)
- **`int_nJobs`** (*`int`*) – The number of thread (default: 1)

Returns

estimator.scores

1D array containing the scores of each genetic features with *float* type

Return type (ndarray)

`genepi.step4_singleGeneEpistasis_Logistic.`**`RandomizedLogisticRegression`**`(`*np_X*, *np_y*`)`

Implementation of the stability selection.

Parameters

- **`np_X`** (*`ndarray`*) – 2D array containing genotype data with *int8* type
- **`np_y`** (*`ndarray`*) – 2D array containing phenotype data with *float* type

doi:10.6342/NTU202000613

Returns

estimator.scores

1D array containing the scores of each genetic features with *float* type

Return type (ndarray)

`genepi.step4_singleGeneEpistasis_Logistic.`**`SingleGeneEpistasisLogistic`**(*str_inputFileName_genotype*, *str_inputFileName_phenotype*, *str_outputFilePath=''*, *int_kOfKFold=2*, *int_nJobs=1*)

A workflow to model a single gene containing two-element combinatorial encoding, stability selection, filtering low quality varaint and L1-regularized Logistic regression with k-fold cross validation.

Parameters

- **`str_inputFileName_genotype`** (*str*) – File name of input genotype data
- **`str_inputFileName_phenotype`** (*str*) – File name of input phenotype data
- **`str_outputFilePath`** (*str*) – File path of output file
- **`int_kOfKFold`** (*int*) – The k for k-fold cross validation (default: 2)
- **`int_nJobs`** (*int*) – The number of thread (default: 1)

Returns

float_f1Score

The F1 score of the model

Return type (float)

2.6.7 step5_crossGeneEpistasis_Lasso

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

`genepi.step5_crossGeneEpistasis_Lasso.`**`CrossGeneEpistasisLasso`**(*str_inputFilePath_feature*, *str_inputFileName_phenotype*, *str_inputFileName_score=''*, *str_outputFilePath=''*, *int_kOfKFold=2*, *int_nJobs=1*)

A workflow to model a cross gene epistasis containing two-element combinatorial encoding, stability selection, filtering low quality varaint and L1-regularized Lasso regression with k-fold cross validation.

Parameters

- **`str_inputFilePath_feature`** (*str*) – File path of input feature files from stage 1 - singleGeneEpistasis
- **`str_inputFileName_phenotype`** (*str*) – File name of input phenotype data
- **`str_inputFileName_score`** (*str*) – File name of input score file from stage 1 - singleGeneEpistasis
- **`str_outputFilePath`** (*str*) – File path of output file
- **`int_kOfKFold`** (*int*) – The k for k-fold cross validation (default: 2)

doi:10.6342/NTU202000613

- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns

tuple containing:

- float_AVG_S_P_train (float): The average of the Peason's and Spearman's correlation of the model for training set
- float_AVG_S_P_test (float): The average of the Peason's and Spearman's correlation of the model for testing set

- Expected Success Response:

```
"step5: Detect cross gene epistasis. DONE!"
```

Return type (tuple)

genepi.step5_crossGeneEpistasis_Lasso.**LassoRegression**(*np_X*, *np_y*, *int_nJobs=1*)
Implementation of the L1-regularized Lasso regression with k-fold cross validation.

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns

float_AVG_S_P

The average of the Peason's and Spearman's correlation of the model

Return type (float)

genepi.step5_crossGeneEpistasis_Lasso.**RegressorModelPersistence**(*np_X*, *np_y*, *str_outputFilePath=''*, *int_nJobs=1*)
Dumping regressor for model persistence

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type
- **str_outputFilePath** (*str*) – File path of output file
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns None

2.6.8 step5_crossGeneEpistasis_Logistic

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

doi:10.6342/NTU202000613

genepi.step5_crossGeneEpistasis_Logistic.**ClassifierModelPersistence**(*np_X, np_y, str_outputFilePath=”, int_nJobs=1*)

Dumping classifier for model persistence

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type
- **str_outputFilePath** (*str*) – File path of output file
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns None

genepi.step5_crossGeneEpistasis_Logistic.**CrossGeneEpistasisLogistic**(*str_inputFilePath_feature, str_inputFileName_phenotype, str_inputFileName_score=”, str_outputFilePath=”, int_kOfKFold=2, int_nJobs=1*)

A workflow to model a cross gene epistasis containing two-element combinatorial encoding, stability selection, filtering low quality varaint and L1-regularized Logistic regression with k-fold cross validation.

Parameters

- **str_inputFilePath_feature** (*str*) – File path of input feature files from stage 1 - singleGeneEpistasis
- **str_inputFileName_phenotype** (*str*) – File name of input phenotype data
- **str_inputFileName_score** (*str*) – File name of input score file from stage 1 - singleGeneEpistasis
- **str_outputFilePath** (*str*) – File path of output file
- **int_kOfKFold** (*int*) – The k for k-fold cross validation (default: 2)
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns

tuple containing:

- float_f1Score_train (float): The F1 score of the model for training set
- float_f1Score_test (float): The F1 score of the model for testing set

- Expected Success Response:

```
"step5: Detect cross gene epistasis. DONE!"
```

Return type (tuple)

genepi.step5_crossGeneEpistasis_Logistic.**GenerateContingencyTable**(*np_genotype, np_phenotype*)

Generating the contingency table for chi-square test.

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type

doi:10.6342/NTU202000613

- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type

Returns

np_contingency

2D array containing the contingency table with *int* type

Return type (ndarray)

`genepi.step5_crossGeneEpistasis_Logistic.`**`LogisticRegressionL1`**`(`*np_X*, *np_y*, *int_nJobs=1*`)`

Implementation of the L1-regularized Logistic regression with k-fold cross validation.

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns

float_f1Score

The F1 score of the model

Return type (float)

`genepi.step5_crossGeneEpistasis_Logistic.`**`PlotPolygenicScore`**`(`*np_X*, *np_y*, *int_kOfKFold=2*, *int_nJobs=1*, *str_outputFilePath=''*`)`

Plot figure for polygenic score, including group distribution and prevalence to PGS

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type
- **int_kOfKFold** (*int*) – The k for k-fold cross validation (default: 2)
- **int_nJobs** (*int*) – The number of thread (default: 1)

Returns None

`genepi.step5_crossGeneEpistasis_Logistic.`**`fsigmoid`**`(`*x*, *a*, *b*`)`

2.6.9 step6_ensembleWithCovariates

Created on Feb 2018

@author: Chester (Yu-Chuan Chang)

`genepi.step6_ensembleWithCovariates.`**`ClassifierModelPersistence`**`(`*np_X*, *np_y*, *str_outputFilePath=''*, *int_nJobs=1*`)`

Dumping ensemble classifier for model persistence

Parameters

- **np_X** (*ndarray*) – 2D array containing genotype data with *int8* type
- **np_y** (*ndarray*) – 2D array containing phenotype data with *float* type

doi:10.6342/NTU202000613

- **`str_outputFilePath`** (*`str`*) – File path of output file
- **`int_nJobs`** (*`int`*) – The number of thread (default: 1)

Returns None

`genepi.step6_ensembleWithCovariates.`**`EnsembleWithCovariatesClassifier`**(*str_inputFileName_feature, str_inputFileName_phenotype, str_outputFilePath='', int_kOfKFold=2, int_nJobs=1*)

A workflow to ensemble genetic features with covariates for L1-regularized Logistic regression.

Parameters

- **`str_inputFilePath_feature`** (*`str`*) – File path of input feature files from stage 2 - crossGeneEpistasis
- **`str_inputFileName_phenotype`** (*`str`*) – File name of input phenotype data
- **`str_outputFilePath`** (*`str`*) – File path of output file
- **`int_kOfKFold`** (*`int`*) – The k for k-fold cross validation (default: 2)
- **`int_nJobs`** (*`int`*) – The number of thread (default: 1)

Returns

tuple containing:

- float_f1Score_train (float): The F1 score of the model for training set
- float_f1Score_test (float): The F1 score of the model for testing set

- Expected Success Response:

```
"step6: Ensemble with covariates. DONE!"
```

Return type (tuple)

`genepi.step6_ensembleWithCovariates.`**`EnsembleWithCovariatesRegressor`**(*str_inputFileName_feature, str_inputFileName_phenotype, str_outputFilePath='', int_kOfKFold=2, int_nJobs=1*)

A workflow to ensemble genetic features with covariates for L1-regularized Lasso regression.

Parameters

- **`str_inputFilePath_feature`** (*`str`*) – File path of input feature files from stage 2 - crossGeneEpistasis
- **`str_inputFileName_phenotype`** (*`str`*) – File name of input phenotype data
- **`str_outputFilePath`** (*`str`*) – File path of output file
- **`int_kOfKFold`** (*`int`*) – The k for k-fold cross validation (default: 2)
- **`int_nJobs`** (*`int`*) – The number of thread (default: 1)

Returns

tuple containing:

- float_AVG_S_P_train (float): The average of the Peason's and Spearman's correlation of the model for training set

doi:10.6342/NTU202000613

- float_AVG_S_P_test (float): The average of the Peason's and Spearman's correlation of the model for testing set
- Expected Success Response:

```
"step6: Ensemble with covariates. DONE!"
```

Return type (tuple)

`genepi.step6_ensembleWithCovariates.`**`LoadDataForEnsemble`**(*str_inputFileName_feature*, *str_inputFileName_phenotype*)

Loading genetic features for ensembling with covariates

Parameters

- **`str_inputFilePath_feature`** (*str*) – File path of input feature files from stage 2 - crossGeneEpistasis
- **`str_inputFileName_phenotype`** (*str*) – File name of input phenotype data

Returns

tuple containing:

- np_genotype (ndarray): 2D array containing genotype data with *int8* type
- np_phenotype (ndarray): 2D array containing phenotype data with *float* type

Return type (tuple)

`genepi.step6_ensembleWithCovariates.`**`RegressorModelPersistence`**(*np_X*, *np_y*, *str_outputFilePath=''*, *int_nJobs=1*)

Dumping ensemble regressor for model persistence

Parameters

- **`np_X`** (*ndarray*) – 2D array containing genotype data with *int8* type
- **`np_y`** (*ndarray*) – 2D array containing phenotype data with *float* type
- **`str_outputFilePath`** (*str*) – File path of output file
- **`int_nJobs`** (*int*) – The number of thread (default: 1)

Returns None

2.7 Release History

All notable changes to this project will be documented in this file.

2.7.1 [2.0.9] - 2019-12-31

Added

- Add standalone AppGenEpi.app

Fixed

- Fix the bug of joblib multithreading

doi:10.6342/NTU202000613

2.7.2 [2.0.8] - 2019-12-26

Fixed

- Fix unsupported image format for ploting PRS

2.7.3 [2.0.7] - 2019-12-26

Fixed

- Fix executable python for AppGenEpi

2.7.4 [2.0.6] - 2019-12-25

Fixed

- Fix file path bug for ploting PRS

Removed

- Remove unused PIL import

2.7.5 [2.0.5] - 2019-12-24

Added

- Add AppGenEpi (GUI)
- Add polygenic risk score calculation for classifier

2.7.6 [2.0.4] - 2019-12-12

Added

- Add sliding windows scanning to deal with mega genes

2.7.7 [2.0.3] - 2019-10-13

Added

- Add GenEpi's documentation to Read the Docs
- Add model persistance for classifier and regressor

2.7.8 [2.0.2] - 2019-09-18

Added

- Add multiprocessing for gene batch running

doi:10.6342/NTU202000613

2.7.9 [2.0.1] - 2019-07-12

Added

- Support self-defiend genome regions (for any species)
- Add argument parser (GenEpi -h)
- Add "GenEpi" to bin when installing
- Add output log
- Add quick test
- Add change log in Changelog.md

Changed

- Change feature encoder for reducing memory usage

Removed

- Remove random forest ensemble

2.7.10 [1.0.3] - 2018-04-22

First preview release

doi:10.6342/NTU202000613

Python Module Index

doi:10.6342/NTU202000613

Index

doi:10.6342/NTU202000613

doi:10.6342/NTU202000613