mathematics

MDPI

Article

BLogic: A Bayesian Model Combination Approach in Logic Regression

Yu-Chung Wei

Graduate Institute of Statistics and Information Science, National Changhua University of Education, No. 1, Jin-De Road, Changhua City 500207, Taiwan; weiyuchung@cc.ncue.edu.tw; Tel.: +886-4-7232105 (ext. 3236)

Abstract: With the increasing complexity and dimensionality of datasets in statistical research, traditional methods of identifying interactions are often more challenging to apply due to the limitations of model assumptions. Logic regression has emerged as an effective tool, leveraging Boolean combinations of binary explanatory variables. However, the prevalent simulated annealing approach in logic regression sometimes faces stability issues. This study introduces the BLogic algorithm, a novel approach that amalgamates multiple runs of simulated annealing on a dataset and synthesizes the results via the Bayesian model combination technique. This algorithm not only facilitates predicting response variables using binary explanatory ones but also offers a score computation for prime implicants, elucidating key variables and their interactions within the data. In simulations with identical parameters, conventional logic regression, when executed with a single instance of simulated annealing, exhibits reduced predictive and interpretative capabilities as soon as the ratio of explanatory variables to sample size surpasses 10. In contrast, the BLogic algorithm maintains its effectiveness until this ratio approaches 50. This underscores its heightened resilience against challenges in high-dimensional settings, especially the large p, small n problem. Moreover, employing real-world data from the UK10K Project, we also showcase the practical performance of the BLogic algorithm.

Keywords: Bayesian model combination; ensemble learning; logic tree; logic regression; machine learning; simulated annealing; UK10K project; variable interactions

MSC: 62C10; 62G08; 62M99; 62R07; 68T09

Citation: Wei, Y.-C. BLogic: A Bayesian Model Combination Approach in Logic Regression. *Mathematics* **2023**, *11*, 4353. https://doi.org/10.3390/math11204353

Academic Editors: Jose Antonio Sáez Muñoz and José Luis Romero Béjar

Received: 28 August 2023
Revised: 10 October 2023
Accepted: 16 October 2023
Published: 19 October 2023

1. Introduction

The development of mathematical and computational models is fundamental in dissecting the intricate nature of relationships within sets of data. Constructing models that describe the relationship between explanatory variables (denoted as X, also called an independent variable, predictor variable, feature, or input) and response variables (denoted as Y, also called a dependent variable, label, or output) has been a continuously evolving topic in the field of mathematics and data science. Whether delving into traditional statistical models, which have been the bedrock of quantitative analysis for centuries, or navigating the waters of the rapidly growing domain of modern machine learning algorithms, researchers and practitioners constantly seek robust methodologies. Many studies, spanning decades and even centuries, aim to establish the relationship between explanatory and response variables based on various theoretical concepts. These models, underpinned by a rich tapestry of mathematical theories, are widely used in many practical fields, from economics to biology, and from physics to social sciences.

The value of these models lies in their versatility. They can be tailored to answer specific questions pertinent to the field of study. For instance, an economist might use such models to gauge the impact of fiscal policy changes on GDP, while a biologist could

employ similar methodologies to explore the relationship between genetic markers and susceptibility to certain diseases [1].

These models, which diligently work to describe the relationship between X and Y, pivot around two core facets: predictability and interpretability. Predictability signifies the model's ability to accurately forecast the response variable given explanatory variables. This capacity to anticipate is not just a theoretical endeavor but is rooted in practical needs. For instance, forecasting stock prices or predicting weather patterns can have tangible economic impacts. Making accurate predictions can lead to saving resources, both monetary and human, and in some cases, such as medical diagnoses, can even save lives.

On the flip side, interpretability delves deeper, seeking to unearth the underlying dynamics, mechanisms, and intricacies that bind the explanatory variable to the response variable. This is not merely about drawing a line of best fit but rather understanding the forces and factors that sculpt this relationship. Understanding the 'why' and 'how' is pivotal. For instance, in a clinical setting, knowing a drug works is essential, but understanding how it works can pave the way for refining its efficacy or reducing side effects [2].

Many modern machine learning models, with their intricate architectures and algorithms, prioritize predictability. They voraciously consume data, sifting through it and teasing out patterns that might elude the human eye. By adopting data-driven approaches, these models can often achieve breathtaking accuracy in their predictions. However, this comes at a cost. The sheer complexity of some of these models, often labeled as "black boxes", can shroud their inner workings, making decisions opaque. This opacity can be a significant impediment, especially in fields such as biomedical research, where understanding the correlation between disease risk factors and the incidence of disease is paramount. Not just predicting, but understanding these correlations can lead to better preventative strategies.

In contrast, traditional statistical models offer a more transparent lens into these relationships. By allowing users to specify relationships and then rigorously testing these assumptions, these models lend themselves to greater scrutiny. The interplay of estimation and hypothesis testing serves as a robust mechanism to assess the significance of each explanatory variable. Some models, particularly tree-based and rule-based ones, are specifically architected to emulate human decision-making processes [3,4]. They set discernible rules for explanatory variables to predict the response variable. While occasionally their predictive accuracy might be eclipsed by machine learning models, their transparency and elucidative prowess often render them more suitable for specific research undertakings.

Recognizing interactions among explanatory variables is pivotal. In the digital age, the ubiquity of big data has transformed the landscape of research. Data sets have ballooned in size, often housing a plethora of explanatory variables. These variables, far from existing in isolation, often entwine in a complex choreography of interactions. Recognizing and understanding these interactions is no longer a luxury but a necessity. Furthermore, fields such as genomic epidemiology stand as a testament to this complexity [1]. Research highlights that certain genetic variations exert substantial influence on disease individually. Conversely, while some variations might not present significant main effects when considered in isolation, their interactive synergy can significantly alter disease outcomes.

In practical decision-making frameworks, there is a prevalent tendency to translate explanatory variables into a binary schema. This approach augments the clarity of discerning how these variables and their synergistic interactions influence the response variable. Illustrative transformations encompass binary explanatory variables (such as smoking status), categorical explanatory variables (for instance, single-nucleotide polymorphism genotypes coded as either dominant or recessive), and continuous explanatory variables (such as determining whether blood pressure exceeds a designated threshold).

In the pursuit of modeling these interaction effects, statistical approaches, with their precision and rigor, offer critical insights. However, these methods frequently demand predefined models, a requirement that becomes daunting when navigating the complex landscape of high-dimensional data. Logic regression (LR) emerges as an invaluable

alternative in such scenarios. Through its use of Boolean combinations of binary explanatory variables, known as a logic tree, LR circumvents the need for presetting interaction types. This nimbleness enhances its interpretability, making it a valuable tool in a researcher's arsenal. Consequently, LR, along with methods derived from its foundational model, has been widely applied to areas emphasizing interpretability, such as medical and genomic topics [5–7], public health and social sciences [8], network systems [9], and robot grasping systems [10]. Beyond its commendable interpretability, it has also been proven to possess exemplary predictive capabilities [11].

However, its reliance on simulated annealing (SA) as an optimization strategy has raised concerns, primarily owing to perceived stability issues. This instability is not just a theoretical concern; it has practical ramifications. Some studies attempt to avoid the instability of SA by using alternative complex solution approaches, while others have pivoted towards ensemble learning methods. Yet, the core issue, the shaky foundation of SA, often remains unaddressed.

Given the aforementioned context, this study is primarily motivated by the necessity of addressing the instability found in simulated annealing within the realm of logic regression. Such instability presents noteworthy challenges, especially considering the crucial role of logic regression in identifying significant interactions among explanatory variables and providing valuable predictive insights. With this understanding, our study pursues two main objectives. Firstly, we aim to illuminate the factors contributing to the instability of simulated annealing within logic regression by leveraging simulation studies. Secondly, we introduce the BLogic algorithm, which incorporates the principles of Bayesian model combination (BMC) to aggregate results from multiple iterations of SA-based logic regression models. This integration aims to mitigate the concerns associated with SA's instability, aspiring to enhance the predictability of a single SA run in logic regression while maintaining the model's prized interpretability.

Subsequent sections of this manuscript have been methodically organized to shepherd readers through our investigation. Section 2 explains the fundamentals of logic regression, describes the logic tree structure, and discusses the use of simulated annealing for optimization. Additionally, we discuss the Bayesian model combination, an ensemble method for integrating multiple models. Section 3 introduces the BLogic algorithm conceived in this study, elucidating its theoretical foundation, detailing its forecasting methods, and showcasing the important scores of interactions. Section 4, supported by simulation studies and experimental data analysis, examines our research objectives, comparing results from individual SA analyses with the combined results of multiple SA iterations merged using the BLogic algorithm. Finally, Section 5 summarizes our findings and suggests possible directions for future research.

2. Preliminaries

In this study, we aim to delve into the potential instability of logic regression when simulated annealing (SA) is used to find the optimal solution. Furthermore, we aspire to strategically employ the concept of Bayesian model combination (BMC) to consolidate the outcomes from numerous logic regression models generated through repeated SA executions, seeking a more steadfast model. Accordingly, the Preliminaries section will expound upon the two central models anchoring our research: logic regression and Bayesian model combination.

2.1. Logic Regression

Logic regression (LR) is a statistical method tailored for analyzing situations where a response variable (Y) is modulated by specific Boolean combinations of binary explanatory variables, denoted as $\{X_1, X_2, \ldots X_p\}$, each taking values of either 0 or 1.

The mathematical formulation characterizing the relationship between the response variable and its predictors is given by $g(E(Y)) = \beta_0 + \sum_{k=1}^{K} \beta_k T_k$. Within this framework, the primary objective of LR is to identify a specific Boolean function, also referred to as a

'logic tree', denoted as T_k for $k = 1, \ldots, K$. These functions encapsulate logical conjunctions of the predictors using operations such as AND, OR, and NOT. For example, a logic tree might be interpreted as (the conjugate of X_5 OR X_3) AND X_1, which denoted as $\left(X_5^c \vee X_3\right) \wedge X_1$. However, these logic trees can be represented in different forms. For uniform representation, Boolean expressions are typically articulated in the Disjunctive Normal Form (DNF) [12]. DNF is fundamentally a series of prime implicants (PIs) connected by OR operations. PIs are either a single explanatory variable or multiple explanatory variables and their conjugates linked through AND operations. The transformation mentioned above is the DNF of a given tree, where both subsets are PIs. For example, $\left(X_5^c \wedge X_1\right) \vee \left(X_3 \wedge X_1\right)$ is the DNF for the tree $\left(X_5^c \vee X_3\right) \wedge X_1$, and the subset of the DNF $\left(X_5^c \wedge X_1\right)$ and $\left(X_3 \wedge X_1\right)$ are PIs. This transformation into PIs offers insight into the complex interactions among specific sets of explanatory variables.

LR is versatile, accommodating a wide range of response variable types, such as continuous, categorical, and even survival outcomes. The bridge between the systematic component and the response is forged through an aptly chosen link function, $g(.)$. Notably, when Y is binary, logic regression can be simplified to a single logic tree T, denoted as $E(Y) = T$. In this study, we initially adopt this more streamlined model, ensuring both clarity and depth in our analyses. We believe that the findings and conclusions drawn from this research can be extrapolated to a broader range of logic regression models.

The process of estimating the regression structure leans heavily on optimization techniques. Specifically, simulated annealing often serves to explore the solution space [13,14], optimizing the configurations of logic trees in LR. SA, derived from the Metropolis–Hastings algorithm, employs a Monte Carlo method. When transitioning between a current solution and a neighboring solution, the decision hinges on their objective value differences and a parameter reminiscent of temperature. If the neighboring solution is superior to the current one, the algorithm shifts to the neighboring solution. Otherwise, a transition probability is set, allowing a potential shift. Initially, a high temperature value is adopted, allowing acceptance of subpar solutions. Over time, as the temperature decreases, the algorithm becomes more stringent, leading to convergence. In LR's landscape, SA assists in navigating the vast potential logic tree configurations. Starting with an initial logic tree, SA refines the combinations, highlighting pivotal predictors and their optimal logical relationships. The gamut of moves employed in this context is well documented in relevant LR literature [15,16].

However, using SA in logic regression presents certain challenges. Studies have shown that SA can sometimes stagnate at local optima within the vast space of logical combinations [15,16]. This convergence can yield suboptimal logic models, potentially misrepresenting underlying data patterns. The intricacies of SA's occasional erratic behavior, particularly when interpreting complex predictor variable interactions such as in single nucleotide polymorphism (SNP) datasets, are further underscored [17]. Additionally, when logic regression models, disrupted by SA's occasional inconsistencies, are integrated into ensemble frameworks such as Logic Forest [18] and LogicFS [19], the overall classifier's performance may decline. While some research, such as MCLR [20], FBLR [17], and GMJMCMC [21], has discussed the instability of SA, suggesting the adoption of Markov chain Monte Carlo methods in lieu of the traditionally used SA, there are challenges. These MCMC-based models are quite intricate, necessitating the setting of parameters to simplify the model upfront. This in turn limits the dimensionality of interactions. Moreover, some models only display results from each iteration during the Monte Carlo process without synthesizing all findings, posing difficulties for practical uses in both prediction and interpretation. Consequently, SA remains the preferred solution for logic regression and its related models.

These insights emphasize the inherent uncertainties in employing SA in logic regression, necessitating thorough exploration and careful management. This understanding is vital as it significantly impacts the predictive and interpretative capacities of models associated with or built upon logic regression.

2.2. Bayesian Model Combination

The paradigm of ensemble learning has significantly shaped the way we approach machine learning problems [22]. Ensemble methods, by definition, aim to consolidate predictions from multiple models to produce a final output that is often more robust and accurate than a prediction from any individual model. Various strategies have been developed to achieve this confluence, broadly categorized under different ensemble learning techniques such as bagging, boosting, and the Bayesian perspective.

Bagging, or bootstrap aggregating, involves generating multiple versions of a predictor by training on subsets of the data [23]. It is based on the principle of leveraging the variance among these different models to produce an aggregated result. Boosting, on the other hand, is an iterative technique that adjusts the weight of an observation based on the last classification [24]. It aims to convert weak individual learners into strong combined learners. Both methods, though distinct, come with their own strengths and challenges.

Contrastingly, Bayesian approaches to model combination, such as Bayesian model averaging (BMA) and Bayesian model combination (BMC), utilize the complete dataset instead of subsets. The Bayesian framework offers a probabilistic mechanism that facilitates the merging of prior knowledge with observed data. In BMA, despite its name suggesting an averaging technique, it behaves more like model selection, emphasizing the identification of the 'best' model [25–27]. BMC, however, truly embodies the essence of model averaging, where each model is weighted based on different strategic considerations [25]. Our research pinpoints a peculiar behavior when simulated annealing is used in logic regression. Due to its inherent stochastic nature, the results of simulated annealing in logic regression manifest instability. Such instability resonates with the notion that relying solely on BMA to discern a single 'optimal' model might not be judicious. Instead, a combination of models through BMC presents a more robust approach.

Delving further into Bayesian model combination, the methodology is anchored on the understanding that it is often more advantageous to amalgamate several models rather than pinpointing the singular best one. Suppose B models have been constructed, denoted as $H = \{h_1, \ldots, h_B\}$. Let $E = \{e_1, \ldots, e_J\}$ represent a spectrum of potential model combinations, wherein an element e_j is perceived as a weight vector for the B models, that is, $e_j = (w_{j1}, \ldots, w_{jB})$ for $j = 1, \ldots, J$. The combination can then be articulated as:

$$p(y_i|\mathbf{x}_i, D, H, E) = \sum_{j=1}^{J} p(y_i|\mathbf{x}_i, H, e_j)\, p(e_j|D) \propto \sum_{j=1}^{J} p(y_i|\mathbf{x}_i, H, e_j)\, p(e_j)\, p(D|e_j) \quad (1)$$

Here, $p(y_i|\mathbf{x}_i, D, H, E)$ represents the probability of predicting Y for the ith individual, conditioned on its corresponding explanatory variable $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, the entire training dataset D, the suite of formulated models H, and the diverse combination strategies encapsulated in E. In cases where the response variable is binary, the category that maximizes this probability is designated.

The predictive probability comprises an ensemble across J combination strategies. For each distinct combination strategy e_j, the predictive probability is essentially the weighted average of the results from the B models, steered by the weight vector $(w_{j1}, \ldots, w_{jB})$ corresponding to e_j. Crucially, the posterior probability of e_j is in direct proportion to the multiplication of its prior $p(e_j)$ and its likelihood $p(D|e_j)$. The formulation of this likelihood function can be predicated upon the predictive accuracy under the guidance of the combination strategy e_j.

The advantages of Bayesian model combinations are numerous. Primarily, it alleviates the instability inherent in individual models. Given that each model might have its own set of strengths and weaknesses, a combination approach ensures that the collective strength is leveraged while minimizing individual model vulnerabilities. Notably, literature substantiates that even rudimentary Bayesian model combination strategies surpass conventional bagging and boosting methodologies and also outperform Bayesian model averaging [25]. Additionally, it furnishes a structured resolution to the conundrum of

model selection, an aspect critically pivotal in instances demanding the utmost model stability and trustworthiness.

Synthesizing the above exposition, the Bayesian model combination paradigm emerges as a robust structure, particularly apt for circumstances marked by model volatility, such as when implementing simulated annealing in logic regression. Thus, subsequent sections of this research not only delve into the latent instabilities associated with the use of simulated annealing in logic regression but also construct a methodology BLogic, inspired by the BMC ethos, to amalgamate multiple logic regression models discovered through multiple runs of simulated annealing, addressing the challenges posed by its inherent instability.

3. Methods

From the aforementioned introduction, it is evident that numerous studies have observed that when simulated annealing (SA) is employed to ascertain optimal solutions for logic regression, instability issues arise. This stands in stark contrast to many methods derived from logic regression, which simply acknowledge this instability without delving into its root causes or attempting to rectify them. To address this gap, our research directly confronts the inherent instability encountered when using SA for logic regression. In response to this challenge, our study adopts the Bayesian model combination (BMC) approach, merging multiple logic regression models that arise from repeated SA methods. The algorithm we have developed, termed BLogic, is anticipated to effectively mitigate the detrimental effects of SA instability on both predictability and interpretability. Furthermore, by amalgamating the recognized significant interactions from each SA-driven logic regression model through BMC, we enhance our ability to pinpoint the most salient explanatory variable interactions in the comprehensive model.

3.1. The BLogic Model Structure

For the sake of simplicity and clarity in this exposition, we primarily use the most elementary architecture of logic regression as an exemplar. This encompasses a binary response variable Y with only a single logic tree T encapsulated within the model, signified by $E(Y) = T$. We posit that the techniques introduced herein can be extended to standard logic regression formats and other response variable types.

Given a dataset of sample size n, each observation contains p binary explanatory variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ and one unique response variable y_i for $i = 1, 2, \ldots, n$. Each observation can be represented as $(\mathbf{x}_i, y_i)$. The entire training dataset can be defined as the set $D = \{(\mathbf{x}_i, y_i): i = 1, 2, \ldots, n\}$. When constructing logic regression using SA, let us assume we repetitively formulate B models, $H = \{h_1, \ldots, h_B\}$. As each model in this context only includes one logic tree, H can be written as $H = \{h_1 = T_1, \ldots, h_B = T_B\}$.

Expanding upon the foundation of these B logic trees (or equivalently, the B models), the BLogic model is constructed by leveraging the concepts of BMC. Analyzing Equation (1), it becomes evident that it is directly tied to the summation across the J combination strategies, namely, $p(y_i \mid \mathbf{x}_i, D, H, E)$. Given the assumption that each combination strategy e_j has a uniform prior, our focus narrows down to determining two primary components within the equation: $p(y_i \mid \mathbf{x}_i, H, e_j)$ and $p(D \mid e_j)$. Moving forward, we interpret the jth combination strategy e_j as a weight vector designated for the B models, denoted as $e_j = (w_{j1}, \ldots, w_{jB})$. Subsequent sections will elucidate the systematic configuration of the combination strategy.

The term $p(y_i \mid \mathbf{x}_i, H, e_j)$ represents the predicted probability of the ith data point y_i, given the collection H of B logic trees generated from training data and the jth weight combination $e_j = (w_{j1}, \ldots, w_{jB})$. As the model used here solely contains one logic tree, each model predicts Y based on the Boolean expressions of its explanatory variables, giving an outcome of either 0 or 1. Hence, $p(y_i \mid \mathbf{x}_i, H, e_j)$ is defined as $w_{j1}\hat{y}_{i1} + w_{j2}\hat{y}_{i2} + \ldots + w_{jB}\hat{y}_{iB}$, where $\hat{y}_{ib}$ is the predicted outcome for model b. In more generic logic regression scenarios, $p(y_i \mid \mathbf{x}_i, H, e_j)$ can be the weighted average of the predicted probabilities from each model.

Additionally, $p(D \mid e_j)$ represents the likelihood function of the training data given the jth combination strategy D. Here, we adopt the commonly assumed "uniform class noise model" in Bayesian model combination strategies [26]. This implies that each instance of training data is independent, and under the combination of e_j, the predictive error rate remains constant at ε_j. Consequently, $p(D|e_j) = \prod_{i=1}^{n} p(\mathbf{x}_i, y_i|e_j) = \varepsilon_j^{n-r_j}(1-\varepsilon_j)^{r_j}$, where r_j signifies the number of correctly predicted samples within the training data under the specific combination strategy e_j.

In addition to the two components previously discussed, the systematic configuration of combination strategies $E = \{e_1, \ldots, e_J\}$ must be considered. The method outlined in the original BMC literature [25] was adopted as the default method to set the combination strategies for the BLogic algorithm. Moreover, the present study offers a clearer explanation of the weight-sampling technique than what is provided in the original literature, incorporating minor adjustments to the sampling method for improved clarity and understanding. It is important to note, however, that the combination strategy methodologies within the algorithm still maintain an open framework. This structure allows users the flexibility to define strategies at their discretion.

The weights for the first q combinations, such as $e_1 = (w_{11}, \ldots, w_{1B})$, $e_2 = (w_{21}, \ldots, w_{2B})$, and so forth up to $e_q = (w_{q1}, \ldots, w_{qB})$, are generated randomly from the Dirichlet distribution with parameters $(\alpha_1, \ldots, \alpha_B)$. We have meticulously set each α_b based on the accuracy of the training data for the bth model, added by one. This ensures that the data's fit to the model is encapsulated in the weight extraction, providing a more rational approach to weight configuration. Subsequently, strategies derived from these q weight combinations are considered.

Following the consideration of strategies derived from the first q weight combinations, we compute the posterior probability of each combination strategy given the data. However, since we assume equal priors for all combination strategies, the posterior probability will be exclusively influenced by the likelihood function discussed in the preceding section. Consequently, we compute the likelihoods $p(D \mid e_j)$, $p(D \mid e_2)$, …, $p(D \mid e_q)$ and identify the combination strategy that maximizes the likelihood function, denoted as $e_{j^*} = (w_{j^*1}, \ldots, w_{j^*B})$.

The weight configurations for the subsequent q combination strategies, i.e., $(e_{q+1}, e_{q+2}, \ldots, e_{2q})$, are then randomly drawn from a Dirichlet distribution characterized by parameters $(w_{j^*1}+1, \ldots, w_{j^*B}+1)$. The generation of new combination strategies ceases either when the likelihood functions calculated across Q consecutive iterations are identical or when a predefined maximum number of iterations is reached. This results in a total of J combination strategies, collectively denoted as $E = \{e_1, \ldots, e_J\}$. After generating these J combination strategies and their respective posterior probabilities, it is imperative to normalize the posterior probabilities. This ensures that the sum of posterior probabilities across the J combination strategies equates to 1.

After establishing all the essential components, Equation (1) can be employed to compute the probability $p(y_i \mid \mathbf{x}_i, D, H, E)$ for an individual data point. In scenarios where the response variable is binary, if this probability equates to or exceeds 0.5, the predicted value of the data point y is assigned as 1, otherwise, it is set to 0. Figure 1 displays a diagram of the algorithm's structure.

3.2. Determining the PI Importance Score within BLogic

Beyond the intricate construction of the BLogic model for response variable prediction, the essence of logic regression is retained, describing the importance of explanatory variables and their interactions. Thus, we introduce the computation of the prime implicant (PI) importance score, enhancing the interpretability of our ensemble model. Here, PI refers to Boolean expressions of the logic tree transmuted into the disjunctive normal form. PIs connote the interactions between features, symbolized by AND operations.

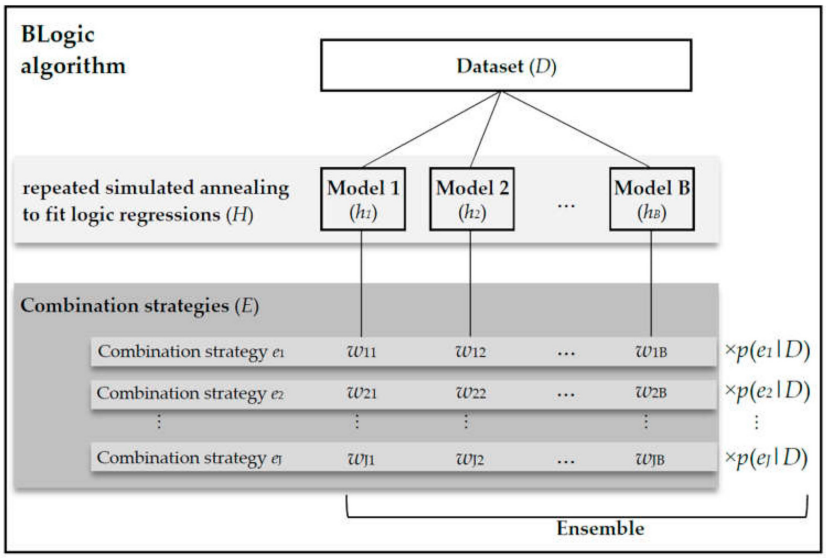


Figure 1. The workflow for the BLogic algorithm.

Suppose a specific PI_l is incorporated into the bth model, also known as the logic tree. In this tree, the importance score of PI_l is denoted as $VIMP_b(PI_l)$. This score is calculated using a permutation-based importance measure. Essentially, all explanatory variables within this PI_l are randomly permuted in the dataset. The difference in accuracy before and after this random permutation signifies the importance score of PI_l. A larger discrepancy in accuracy between pre-permutation and post-permutation implies a higher importance of the PI_l.

Integrating results from B trees and J combination strategies, the importance score for a specific PI_l in BLogic is formulated using the predetermined J combination strategies, alongside their normalized posterior probabilities. This is mathematically represented as:

$$VIMP.BLogic(PI_l) = \sum_{j=1}^{J} \sum_{b:PI_l \in b} VIMP_b(PI_l) \times w_{jb} \times p(e_j|D) \quad (2)$$

After determining the importance scores for all PIs included in the BLogic model, a comprehensive bar chart can be generated. This visual representation effectively highlights the critical interactions.

4. Results

4.1. Simulation

In this analysis, we employ simulated data as a foundation to illuminate the potential instabilities of logic regression when using the simulated annealing (SA) method. Furthermore, we highlight the efficacy of our BLogic algorithm. This approach consolidates multiple logic regression models generated via SA, underscoring its prowess in both predictive and explanatory capacities. Consequently, our benchmarking primarily juxtaposes foundational logic regression with our novel BLogic.

Although numerous established methods have evolved from logic regression, we specifically chose not to compare them in our study, focusing instead only on the fundamental and original logic regression. One primary reason is that MCMC-based approaches do not employ SA, and the instability of SA is one of the main issues we aim to address. Additionally, due to the complexity of these methods, there is a need to preset parameters to simplify the model, creating a different comparison baseline. Most importantly, some of these methods only provide the outcomes of each MCMC iteration and lack a comprehensive strategy to amalgamate these results for practical prediction and vital interaction

interpretation. Consequently, comparing them with our method in terms of predictability and interpretability becomes problematic.

Furthermore, certain methodologies, such as LogicFS and Logic Forest, blend ensemble learning with logic regression. These techniques leverage the bootstrap aggregation (bagging) method to amalgamate several logic regression models. However, their modeling is grounded in bootstrap samples, not the complete dataset. Moreover, they recognize the instability inherent in SA without addressing or amending it. Their primary goals do not align with ours. Given that models born from the bagging process usually employ in-bag and out-of-bag validation methods, and considering our study does not harness the bagging approach, our evaluation criteria differ. Hence, these methods were set aside in our benchmarking.

The subsequent subsections are structured as follows: Section 4.1.1 provides details on the parameter configurations for the simulated data. Section 4.1.2 explores potential factors leading to instability in the SA technique when applied to logic regression. Section 4.1.3 delves into the BLogic algorithm's method of amalgamating multiple logic regression models derived from SA, emphasizing its proficiency in both predictability and interpretability.

4.1.1. Parameter Settings for Simulated Data

To investigate the impact of data composition characteristics on model performance, design parameters for simulations were set. Two total sample sizes, n, of 200 and 1000 were considered. For both sizes, samples included individuals designated as $y = 1$ (cases) and $y = 0$ (controls). A case-to-control ratio in two configurations for each sample size was established: 1:1 and 1:2. Recognizing the vital interplay between the number of predictors and the sample size, the number of predictors, p, was set at eleven relative levels relative to n: $0.1n$, $0.25n$, $0.5n$, $1n$, $2.5n$, $5n$, $10n$, $25n$, $50n$, $100n$, and $250n$.

In alignment with the literature [18], explanatory variables and their corresponding response variables were generated for every scenario. It is important to mention that each dataset was designed to contain a single true prime implicant (PI) that represented the interaction affecting the response variable. These PIs ranged from two-way to eight-way interactions among the explanatory variables, with the response variable being determined through a Boolean operation on the PI. Each explanatory variable was distributed independently and identically, adhering to a Bernoulli distribution. The parameters for this distribution were determined based on the given n, p, and PI setups. For a thorough subsequent analysis, we generated 100 training datasets and an independent testing dataset for each parameter combination.

For illustration, let us consider a scenario where there is a true two-way interaction serving as the PI. In this case, X_1 and X_2, which constitute the true PI, are independently generated through a Bernoulli distribution with identical parameter values. The process of generation produces a number of samples that surpasses the initially set sample size. From this extended pool, samples for case and control groups are selected based on the predetermined size and case-to-control ratio requirements. Samples in which (X_1 AND X_2) equal 1 are randomly selected until the count meets the number previously established for the case group ($y = 1$). In a similar fashion, samples where (X_1 AND X_2) equal 0 are randomly chosen to reach the predetermined count for the control group ($y = 0$). After this careful selection, values of X_1 and X_2 for the necessary samples are ascertained. Subsequently, additional explanatory variables, such as X_3, X_4, ..., X_p, which are not part of the PI, are generated for each sample. These additional variables are independently drawn from a Bernoulli distribution with a parameter of 0.5, ensuring alignment with the parameters for n, p, and PI previously specified for the ensuing analysis.

4.1.2. Instabilities in Logic Regression via Simulated Annealing

This subsection focuses on exploring the intrinsic data attributes that might lead to simulated annealing instabilities in logic regression. We bypass discussions on SA hyperparameters, including the initial temperature ($\text{Temp}_{\text{start}}$), the final temperature (Temp_{end}),

and the iteration count ($iter_{SA}$). It should be noted, however, that we posit that the choice of these hyperparameters in SA could, to some degree, impact the stability of logic regression. Still, compared to the properties of the dataset, this is likely a minor effect. Moreover, general users often adhere to the default settings provided by the package, making repeated adjustments to hyperparameters and re-executing SA impractical. For our analysis, the SA hyperparameters were $Temp_{start}$ = 100, $Temp_{end}$ = 0.1, and $iter_{SA}$ = 50,000 [28]. Moreover, we set an upper limit of leaves = 8 for the number of leaves in the logic tree, in line with the original recommendations for logic regression [15].

In examining SA's potential instabilities, each of the 100 training datasets underwent 100 iterations of SA-based logic regression, with the mean performance across iterations subsequently evaluated. Though initial simulations considered both n = 200 and 1000, analogous trends appeared for both. As would be expected, the results for larger samples (n = 1000) were predictably more stable, making it challenging to clearly investigate the instability trends associated with simulated annealing. Therefore, we limit our presentation to the results for n = 200. Additionally, both 1:1 and 1:2 case-to-control ratios were explored. The observations indicate analogous trends, though the 1:2 ratio slightly underperformed. We thus center our discussion on the equal-proportion scenario. Initially, both two-ways to eight-ways true interactions were considered. As anticipated, the performance for lower-ways surpassed that of higher-ways. Therefore, we present the two-ways results in the manuscript.

Regarding predictive performance, Figure 2a,b depict training and testing dataset accuracy for logic regression, represented by gray dots and lines, respectively. These figures illustrate that both training and testing performances decline and become notably more varied as the number of explanatory variables increases, particularly when surpassing the sample size n. Given that logic regression is a subset of statistical regression techniques, regression methods might lack unique solutions or struggle to find optimal ones when the number of explanatory variables (or parameters to estimate) exceeds the sample size. This challenge could contribute to inconsistent model outcomes and reduced predictive capabilities with SA. Notably, when the number of explanatory variables substantially exceeds the sample size, the accuracy for the training dataset can outperform the testing dataset by 10–20%. This potential overfitting aligns with documented challenges when using logic regression for prediction in certain contexts [16]. The F1-scores for the training and testing datasets follow trends akin to the accuracy measures and as such have been omitted from the figures to maintain focus and conciseness in the presentation of results.

In the context of model interpretability, Figure 3a scrutinizes the capability of the model to correctly identify true PIs through 100 repeated iterations of simulated annealing in logic regression. Our analysis reveals that when the count of explanatory variables is either equal to or less than the sample size, a substantial majority—exceeding 90%—of the models generated via these 100 SA iterations are successful in pinpointing the true PIs. However, this rate of successful identification undergoes a steep decline as the number of explanatory variables starts to outnumber the sample size. For instance, when the number of explanatory variables is tenfold of the sample size, the mean detection rate drops precipitously to 48.13%. Moreover, when the ratio of explanatory variables to sample size scales between 100 and 250, discerning genuine interactions becomes exceptionally difficult.

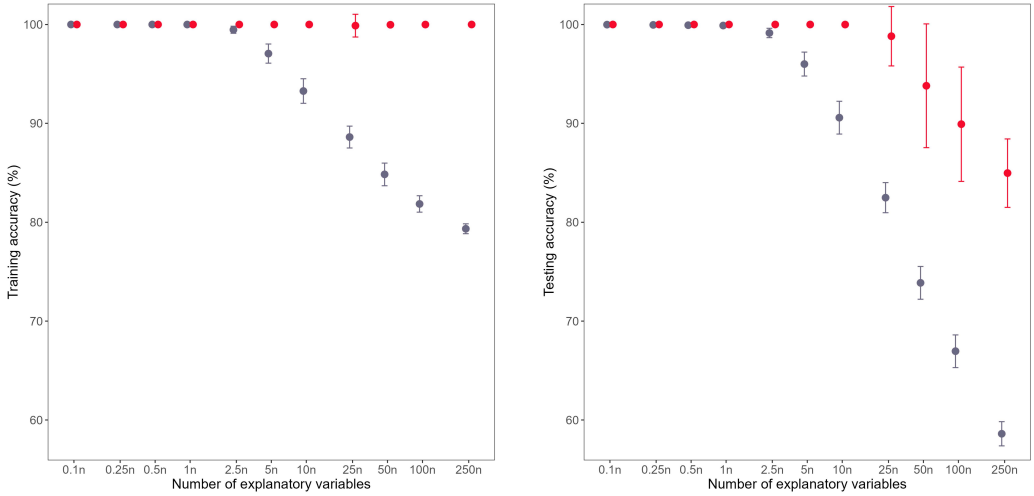


Figure 2. Predictability assessment of logic regression (represented by gray dots and lines) and BLogic algorithm (represented by red dots and lines). The *X*-axis of each panel displays the number of explanatory variables on a logarithmic scale. Each bar displays the mean ± 1 standard deviation. (**a**) Training accuracy; (**b**) Testing accuracy.

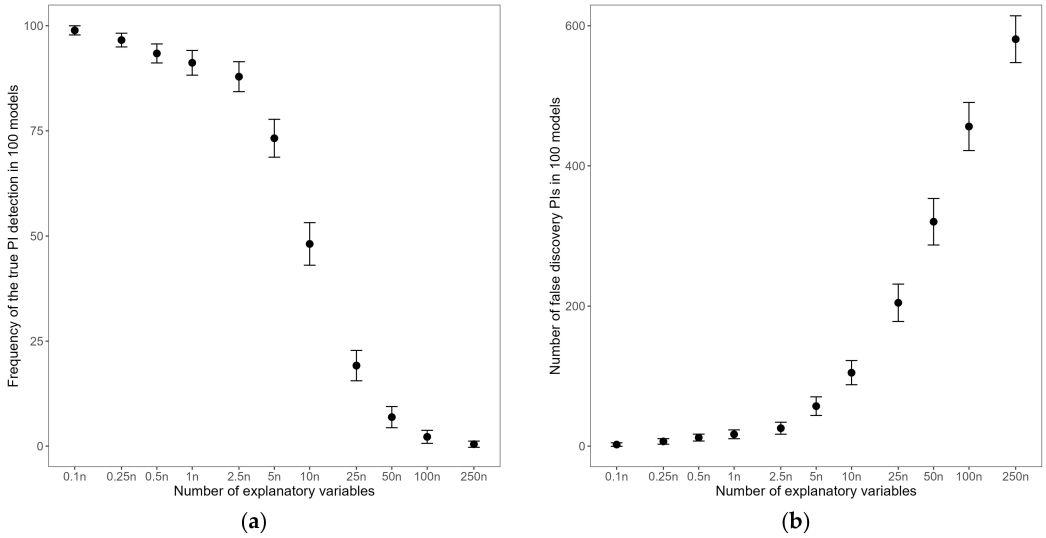


Figure 3. Analysis of instability in logic regression via repeated simulated annealing. The *X*-axis of each panel displays the number of explanatory variables on a logarithmic scale. Each bar displays the mean ± 1 standard deviation. (**a**) Frequency of the true PI detection; (**b**) Number of false discovery PIs.

These observations highlight the inherent limitations of individual SA runs in uncovering true PIs, especially in cases where the predictor variables vastly outnumber the samples. They also shed light on the potential shortcomings of more complex ensemble techniques, such as Logic Forest. Despite its aggregation of multiple models generated from bootstrap samples, Logic Forest might still face challenges in identifying true interactions due to the constraints of individual SA implementations within each model.

Furthermore, Figure 3b denotes the number of detected PIs during the 100 SA iterations, excluding the true PI. These additional PIs can be regarded as false discoveries. The graph reveals that as the number of explanatory variables significantly surpasses the sample size, each SA iteration might detect varying PIs. This variability reaffirms the instability of SA executions.

From a computational standpoint, our analyses were conducted on the Taiwania 1 supercomputer, hosted by the National Center for High-performance Computing within the National Applied Research Laboratories in Taiwan. The machine is equipped with dual Intel Xeon Gold 6148 2.40 GHz CPUs and offers configurations of either 192 GB or 384 GB of memory. We noted a discernible increase in computational time as the number of explanatory variables expanded. When the number of explanatory variables reached the size of the sample, the average time required was approximately 0.19 s. As the number of explanatory variables ranged from 2.5 to 25 times the sample size, computation times varied between 0.22 and 0.5 s. When the number of explanatory variables reached 50 and 100 times the sample size, the average time rose to around one second. This climbed sharply to an average of approximately 5.08 s when the number of explanatory variables was 250 times the sample size.

In conclusion, the ratio of explanatory variables to samples is a decisive factor impacting the efficacy of SA in identifying optimal logic regression solutions. The performance degrades rapidly when the number of explanatory variables surpasses the sample size, influencing predictability, interpretability, and computation times. While other factors might subtly impact performance, such as the complexity of the explanatory variable structure influencing response variables, the overall results are consistently poorer in more intricate scenarios.

4.1.3. Performance of BLogic Algorithm

From the previous subsection, it was established that the performance of logic regression constructed by simulated annealing can be unstable under certain data characteristics, leading to unsatisfactory predictive and interpretative outcomes. In this section, we delve into simulated data with the same settings to examine how the BLogic algorithm, by merging multiple logic regression models obtained through repeated simulated annealing via a Bayesian model combination (BMC) approach, can enhance the performance of a single logic regression constructed by SA both in prediction and interpretation.

In the BLogic algorithm, each data point undergoes repeated simulated annealing to obtain 100 logic tree models (i.e., $B = 100$). Throughout the iterations where a Dirichlet distribution sampling determines the weights of combination strategies, ten combination strategies are sampled in each iteration (i.e., $q = 10$). If the maximum prediction accuracy remains consistent over three consecutive iterations or when the iteration count reaches its threshold (Q = 10,000), the algorithm halts its generation of further combination strategies. The final ensemble consists of 100 models pinpointed by SA, integrated with the sampled combination strategies by the BLogic algorithm for predicting and pinpointing vital PI.

Regarding predictive performance, the red dots and lines in Figure 2a demonstrate that BLogic delivers remarkable results in terms of the accuracy of the training dataset, regardless of the ratio of samples to explanatory variables. This performance might be attributed to the fact that BLogic's design integrates the training dataset's performance over a range of combination strategies, thus shaping a likelihood function and, furthermore, the posterior probability. For the testing dataset, BLogic sustains commendable performance even when the number of explanatory variables exceeds the sample size. This distinction is particularly evident when comparing the red dots and lines representing BLogic in Figure 2b to the gray dots and lines representing logic regression. With the explanatory variables being ten times the number of samples, BLogic's testing accuracy almost invariably nears a remarkable 100%. Such prowess markedly surpasses a solitary instance of SA logic regression, which averages around 90%. Furthermore, as the tally of explanatory variables skyrockets to a staggering 250 times the sample size, BLogic achieves a testing accuracy of 84.97%, dramatically outperforming the 58.60% mean accuracy garnered from a standalone SA-based logic regression. The F1-scores for the training and testing sets still show similar trends to accuracy metrics, so we have left them out of the figures for clarity and brevity.

For interpretability, our examination utilizes two metrics. The first metric is the "Ranking of the true PI detected by BLogic". Considering that each dataset contains

only a single true PI, an average rank close to 1 indicates the true PI's correct detection. Insights from Figure 4a delineate that the true PI's consistent detection and its crowning rank as the most paramount are evident when the number of explanatory variables is equal to or less than 25 times the sample size. However, as this ratio escalates to 50, 100, or even 250, the true PI's importance ranking may experience fluctuations, occasionally missing the top berth but on average landing within the top two or three positions. Hence, when the number of explanatory variables significantly outnumbers the sample size, it is recommended to observe multiple top-ranked PIs based on their importance scores.

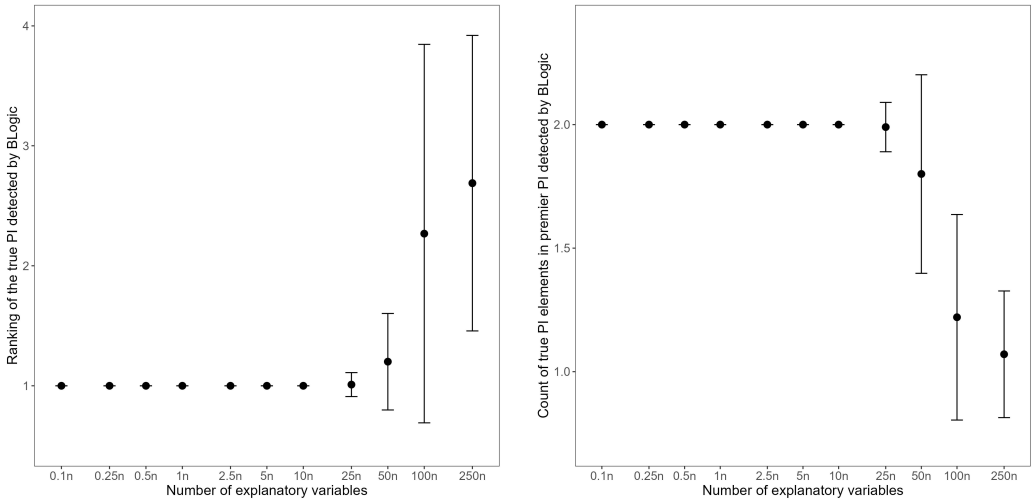


Figure 4. Interpretability assessment of the BLogic algorithm. The *X*-axis of each panel represents the number of explanatory variables on a logarithmic scale. Each bar displays the mean $\pm$ 1 standard deviation. (**a**) Ranking of the true PI; (**b**) Count of true PI elements in premier PI.

On the other hand, since we have only set a single true PI, if the model boasts great interpretability, the premier PI detected by BLogic should ideally be the true PI. Yet, in scenarios where there is an abundance of explanatory variables or when the true PI encompasses complex higher-way interactions, it may become challenging for the premier PI to fully capture all the variables within the true PI. Nonetheless, it remains imperative that the detected premier PI at least embodies elements of the true PI instead of being entirely disparate. To quantify this nuance, we introduced a metric termed the "count of true PI elements in premier PI detected by BLogic." For illustration, if the true PI was a two-way interaction $(X_1 \wedge X_3)$, and BLogic's premier PI embraces both X_1 & X_3, the metric equals 2. If it contains only one of them, the value is 1, and if neither, the value is 0. Observations from Figure 4b indicate that when the number of explanatory variables is less than or up to roughly 25 times the sample size, BLogic's premier PI always encompasses elements of the true PI. However, as this ratio increases, the detected premier PI may begin to incorporate other non-critical explanatory variables. When the count of explanatory variables hits 250 times the sample size, the premier PI might only contain one element of the true PI. The recommendations based on Figure 4 indicate that when the number of explanatory variables significantly outweighs the sample size, the premier PI may comprise some elements of the true PI but may not entirely represent it. Observing multiple top-ranked PIs based on their importance scores is suggested.

4.2. Experimental Data

To assess the capabilities of our developed BLogic algorithm, we drew upon next-generation sequencing data from the UK10K Project [29], housed within the European Genome-Phenome Archive. These data encompass both patients with specific diseases and healthy control samples, allowing us to employ a case-control study approach. From the

provided sequencing data, we underwent a series of preprocessing steps to extract SNPs, which then served as explanatory variables for our model. The following subsections delve into the specifics of the data and preprocessing techniques and showcase the comparative analytical outcomes between the BLogic algorithm and logic regression.

4.2.1. Data Overview and Preprocessing

To evaluate the model, data from the UK10K Project was curated and organized into a case-control study design, with cases and controls delineated as binary response variables. Ninety-seven individuals diagnosed with severe insulin resistance (SIR), a rare condition, were chosen as the case group. To balance data between the case and control cohorts, 100 control samples were drawn from the TwinsUK subset within the UK10K Project, considering factors such as dizygotic twinning, sequencing quality, and depth.

All selected samples underwent preprocessing to identify uniform genomic variations as explanatory variables. Since the UK10K Project utilized whole-genome sequencing for controls and exome sequencing for cases, the analysis focused on target regions defined by exome sequencing. Following preprocessing guidelines set by tools such as Samtools [30] and the Genome Analysis Toolkit (GATK) [31], SNPs were extracted for consideration. Due to model constraints limiting the number of explanatory variables, only SNPs from a single chromosome were used for subsequent analysis. Specifically, 28,144 SNPs from chromosome 19 were chosen, as this chromosome is recognized for containing genes associated with the genomic mechanisms of SIR, as evidenced by scientific research [32–40].

Furthermore, based on typical data selection criteria in genome-wide association studies [41], we chose the final samples and SNPs for the model. These criteria included a minor allele frequency of $\geq$0.01, passing the Hardy–Weinberg Equilibrium test with a p-value $> 5.7 \times 10^{-7}$, and an identical-by-state value of less than 86%. Population stratification issues were overlooked as all samples were from the UK population. After exclusions, the dataset comprised 86 SIR patients and 100 control samples. Each SNP was converted into two binary dummy variables, signifying dominant and recessive effects. After removing non-informative features, 21,387 features were left for further analysis.

4.2.2. Analysis Results

To thoroughly assess the predictive performance of our proposed BLogic algorithm in comparison to logic regression with single simulated annealing, we employed a repeated 10-fold cross-validation, conducted 50 times. Common hyperparameters for both BLogic and logic regression were uniformly set, encompassing SA parameters ($\text{Temp}_{\text{start}} = 100$, $\text{Temp}_{\text{end}} = 0.1$, and $\text{iter}_{\text{SA}} = 50{,}000$) and a maximum of 8 leaves for the logic tree. For BLogic-specific hyperparameters, each data batch was configured to undergo SA 100 times ($B = 100$), sampling ten combination strategies ($q = 10$) during each cycle. The algorithm ceases operation either when maximum prediction accuracy remains stable across three successive iterations or upon reaching its predetermined threshold ($Q = 10{,}000$).

After obtaining the accuracy and F1-score for each of the 10-fold cross-validations, the results from the fifty repetitions were averaged to compute the mean and standard deviation (sd) to gauge the predictive performance of the BLogic algorithm against the single SA in logic regression. These outcomes are detailed in Table 1. The BLogic algorithm substantially surpasses logic regression in both mean accuracy and F1-score. Additionally, the standard deviations underscore the enhanced stability of BLogic, further distinguishing it from logic regression in terms of consistency in performance. With regards to the training set, although the mean accuracy and F1-score of BLogic are only marginally superior to those of logic regression, this subtle edge might be due to the effective fitting of the logic regression model to the training data. However, a closer examination of the standard deviation for both metrics unequivocally demonstrates that BLogic consistently maintains a higher level of stability compared to its counterpart. These predictive results align with the findings from the simulation

Table 1. Prediction performance on experimental data.

		Training Set		Testing Set	
	(%)	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
BLogic	mean	100	100	99.88	99.85
	(sd)	(0)	(0)	(0.22)	(0.32)
logic regression	mean	99.84	99.82	95.17	94.36
	(sd)	(0.10)	(0.11)	(0.60)	(0.89)

On the interpretive front, to illustrate the crucial SNPs and their interactions identified by BLogic within this dataset, the unpartitioned data was processed again using the BLogic algorithm, keeping the hyperparameter settings consistent with those previously mentioned. This process culminated in the generation of 30 combination strategies, with iterations ceasing when no further enhancements in predictive accuracy were observed. Additionally, we calculated the PI importance scores, as mentioned earlier, and highlighted the most significant PIs in descending order based on these scores. It is worth noting that the results of the PIs detected during individual runs of SA in logic regression are not provided here. This decision was made due to the inconsistent PI outcomes from each individual SA run in logic regression, deeming them unsuitable for presentation.

Figure 5 displays the top 10 PIs obtained from the BLogic algorithm. In the PI notation, an '!' prefix to the SNP rs number signifies a complement set, while the suffixes '_1' and '_2' indicate dominant and recessive coding, respectively. The top-ranking PIs often involve interactions between two or three SNPs. A search using the Genome Data Viewer at the National Center for Biotechnology Information revealed that all SNPs included in the top 10 PIs are located within the genomic bands of 19p13 and 19q13. Numerous studies have pinpointed gene mutations within the 19p13 region that influence the insulin receptor, subsequently leading to insulin resistance [32–36]. Additionally, some research confirms that genetic variants on 19q13 can cause severe insulin resistance [37,38], as well as Type 2 diabetes mellitus associated with insulin abnormalities [39,40]. The PI with the highest importance score identified by BLogic, which has a score notably higher than other PIs, involves an interaction between the SNPs rs162124 (located at 19q13.41) and rs11085209 (located at 19p13.2). Experts are thus recommended to not only examine the regions of 19p13 and 19q13 independently but also to further probe into the potential impacts on the SIR mechanism pathway resulting from mutations within these regions. Moreover, within the top 10 PIs, the SNPs rs11085209 (located at 19p13.2) and rs10415889 (located at 19p13.11) frequently interact with other SNPs. Hence, this outcome suggests that experts might consider a more in-depth investigation of the genetic variants within the 19p13 sub-bands, particularly 19p13.2 and 19p13.11, and their impact on the SIR mechanism.

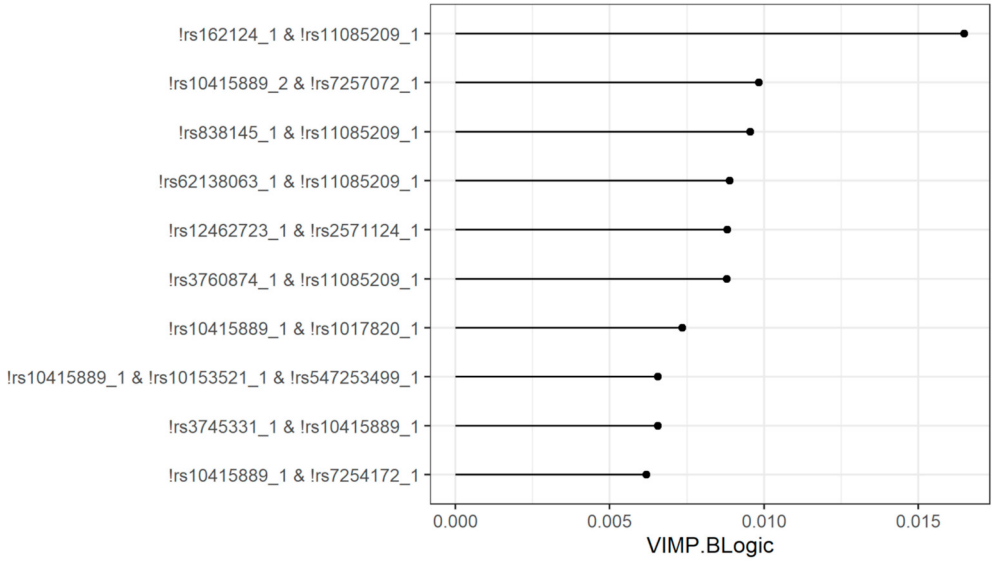


Figure 5. Top 10 prime implicants by BLogic from the UK10K project on severe insulin resistance.

5. Conclusions and Discussions

Logic regression presents a unified model that utilizes Boolean combinations of binary explanatory variables to predict response variables. This structure inherently identifies crucial interactions among the explanatory variables. It is in stark contrast to conventional statistical methods, which necessitate predefined interaction categories. Thus, it is particularly advantageous when seeking to uncover significant interactions among many explanatory variables.

Simulated annealing (SA) is commonly adopted to find the optimal solution in logic regression. Numerous studies have acknowledged the instability of SA in this context, but the underlying causes of this instability remain largely unexplored. In our research, we specifically employed simulated data to probe the characteristics underlying the instability of SA in logic regression. In our simulation studies, when only one two-way interaction PI is set in the data, we found that the challenges arising from SA's instability begin to manifest when the number of explanatory variables is more than ten times the sample size. Furthermore, when the quantity of explanatory variables greatly exceeds the sample size, the instability in SA becomes profoundly evident. This results in a significant drop in predictive precision and a failure to pinpoint vital interactions. We believe that when the data contain more complex interactions, the performance of the model might be adversely affected by SA's instability, even when the number of explanatory variables does not greatly exceed the sample size. Such instability undermines the unique advantage of logic regression: its inherent capability to autonomously detect interactions.

To address this issue, our study proposed the BLogic algorithm. This method relies on repeatedly employing SA to construct various logic regression models on the same dataset. The Bayesian model combination (BMC) approach, suitable for assimilating multiple unstable models, is then employed to combine each logic regression using different combination strategies. This methodology has demonstrated superiority in prediction accuracy compared to relying solely on a single unstable result from SA in logic regression. Furthermore, through this theoretical method, we can systematically consolidate all interactions identified by multiple SAs and evaluate the relative importance of each interaction. Moreover, the influence of the ratio between explanatory variables and sample size is attenuated in this approach. In the setting of our simulation study, it is only when the number of explanatory variables exceeds 50 times the sample size that prediction and interpretability begin to show some minor effects.

Based on our research findings, there is substantial scope for further investigation. One noteworthy area of interest is determining the optimal number of logic regression

models to be constructed within BLogic, specifically deciding on the appropriate setting for the hyperparameter B. While we highlighted findings with $B = 100$ in our simulations, additional tests with $B = 200$, 300, 400, and 500 ($n = 200$, case-to-control ratios 1:1, and two-way true interactions) have also been conducted. The results are presented in Tables 2 and 3, focusing on predictability and interpretability, respectively. Even though the patterns in predictability and interpretability remained relatively stable across varying B values, larger B values seemed to slightly better mitigate the issues arising from a high ratio of explanatory variables to sample size. While there was no significant difference in performance across our chosen B values ranging from 100 to 500, the selection of B still might influence the analysis results. Opting for a smaller B might not fully capture the breadth of potential SA outcomes, potentially making the merging of unstable SA instances ineffective. Conversely, a larger B, while encompassing varied SA results, would demand more computational resources due to repeated SA runs and the necessity to determine weights for each model within the BLogic algorithm. Additionally, in scenarios where the sample size significantly outnumbers the explanatory variables, as demonstrated in our simulations, SA often generates nearly identical logic tree outputs. As a result, executing SA numerous times might be inefficient. For the time being, we have set a default value, drawing inspiration from the commonly used default value of 100 in random forests [42]. Future work might consider adjusting the value of B based on factors such as the total number of explanatory variables in the data, the number of leaves in each logic tree, the number of potential key explanatory variables, and the order of interactions. Subsequent investigations could seek to methodically understand the interconnectedness of these considerations. As an example, with a predetermined number of explanatory variables, pinpointing the lowest B value essential to identifying a specific sequence of PI might be of interest. Employing simulation studies could also be valuable in analyzing the patterns and relations of these parameters.

Table 2. Impact of hyperparameter B on the predictability of the BLogic algorithm.

	B		Number of Explanatory Variables										
			0.1*n*	**0.25*n***	**0.5*n***	**1*n***	**2.5*n***	**5*n***	**10*n***	**25*n***	**50*n***	**100*n***	**250*n***
Training accuracy (%)	100	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	99.89 (1.15)	99.98 (0.21)	99.99 (0.05)	100 (0)
	200	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	99.87 (1.20)	99.99 (0.07)	100 (0)	100 (0)
	300	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	99.70 (1.79)	99.96 (0.31)	100 (0)	100 (0)
	400	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	99.65 (1.88)	99.995 (0.05)	100 (0)	100 (0)
	500	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	99.38 (2.61)	100 (0.05)	100 (0)	100 (0)
Testing accuracy (%)	100	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	99.995 (0.05)	98.81 (3.01)	93.80 (6.26)	89.91 (5.78)	84.97 (3.46)
	200	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	96.64 (5.24)	90.54 (8.32)	89.61 (7.19)	86.14 (3.33)
	300	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	95.20 (6.73)	90.69 (9.45)	89.55 (7.64)	86.96 (3.34)
	400	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	93.88 (8.07)	89.71 (9.71)	89.15 (7.95)	86.99 (3.49)
	500	Mean (sd)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	93.57 (8.48)	89.74 (9.82)	89.14 (7.62)	87.12 (3.65)

Table 3. Impact of hyperparameter B on the interpretability of the BLogic algorithm.

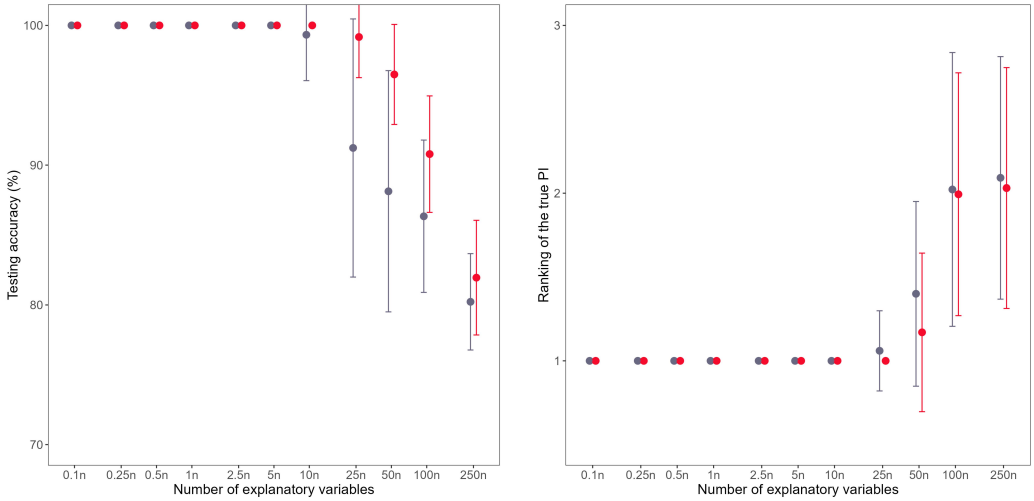
	B		Number of Explanatory Variables										
			$0.1n$	$0.25n$	$0.5n$	$1n$	$2.5n$	$5n$	$10n$	$25n$	$50n$	$100n$	$250n$
Ranking of the true PI detected by BLogic	100	Mean	1	1	1	1	1	1	1	1.01	1.20	2.27	2.69
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.10)	(0.40)	(1.58)	(1.23)
	200	Mean	1	1	1	1	1	1	1	1.01	1.28	1.87	3.00
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.10)	(0.45)	(0.56)	(1.76)
	300	Mean	1	1	1	1	1	1	1	1	1.29	1.96	2.99
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.46)	(0.55)	(1.93)
	400	Mean	1	1	1	1	1	1	1	1.01	1.22	1.98	2.85
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.10)	(0.42)	(0.51)	(1.22)
	500	Mean	1	1	1	1	1	1	1	1.02	1.25	1.99	3.34
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.14)	(0.43)	(0.48)	(2.68)
Count of true PI elements in premier PI detected by BLogic	100	Mean	2	2	2	2	2	2	2	1.99	1.80	1.22	1.07
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.10)	(0.40)	(0.42)	(0.26)
	200	Mean	2	2	2	2	2	2	2	1.99	1.72	1.23	1.03
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.10)	(0.45)	(0.42)	(0.17)
	300	Mean	2	2	2	2	2	2	2	2	1.71	1.17	1.01
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.46)	(0.38)	(0.10)
	400	Mean	2	2	2	2	2	2	2	1.99	1.78	1.14	1.01
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.10)	(0.42)	(0.35)	(0.10)
	500	Mean	2	2	2	2	2	2	2	1.98	1.75	1.12	1.01
		(sd)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0.14)	(0.44)	(0.33)	(0.10)

Secondly, overfitting remains a concern. Previous literature, along with the gray lines and dots in Figure 2, suggests that a single logic regression model can overfit under certain data conditions. The red lines and dots in the figure indicate that while BLogic has marginally better training accuracy than testing, it does not amplify the overfitting problem. This discrepancy in BLogic's training and testing accuracy may arise from the inherent instability of a logic regression run with a single SA. A potential solution for future research might involve the application of cross-validation, segmenting the full dataset into training and validation sets. The training set could be used to develop the logic regression model with SA, and the validation set could help determine the likelihood function and subsequent posterior probabilities for each combination strategy. This approach might mitigate overfitting by decreasing the contribution of the training set. However, such a modification would necessitate a thorough re-evaluation of the theoretical framework of the BMC model due to the dataset division.

Thirdly, the BLogic algorithm adopts the method outlined in the original BMC literature as the default approach for configuring combination strategies within BMC. This method uses the performance of individual combination strategies as a basis to update the parameters within the Dirichlet distribution, continually iterating to generate a series of combination strategies. While theoretically plausible, there are reservations cautiously acknowledged regarding its consistent ability to yield optimal combination strategies. The effectiveness of this method might vary due to differences in data or potential correlations with other hyperparameters within the model. The optimal strategies of model combination for BLogic remain an open area of research and require further study.

A straightforward comparison is presented with a naive approach, which assigns equal weights to each logic regression model obtained from a single SA run and the combination strategies deployed by the BLogic algorithm. This comparison, generated with true two-way PIs, a sample size of 200, and a 1:1 case-to-control ratio, deliberately selects a scenario with a reduced number of SA runs, namely 50, to spotlight the effective performance of the BMC technique. To keep the article's focus sharp, Figure 6a represents predictability

solely through testing accuracy, while Figure 6b illustrates the ranking of true PI, shedding light on their interpretability. In most scenarios, the combination method of BLogic's model (denoted by red dots and lines) not only consistently secures higher accuracy but also assigns the highest importance scores to the true PI, thereby correctly identifying it as a priority, compared to the equal weight approach (signified by gray dots and lines). This illustration emphasizes the effectiveness of employing the BMC technique for combinations. However, it is crucial to acknowledge that alternative combination strategies necessitate further and more detailed exploration.



(**a**) (**b**)

Figure 6. Performance assessment of equal weight combination (represented by gray dots and lines) and BLogic via BMC default combination strategy (represented by red dots and lines). The *X*-axis of each panel displays the number of explanatory variables on a logarithmic scale. Each bar displays the mean $\pm$ 1 standard deviation. (**a**) Testing accuracy; (**b**) Ranking of the true PI.

Lastly, a significant contribution of our research lies in the revelation through simulated studies that SA outcomes become notably unstable when the number of explanatory variables greatly exceeds the sample size. This "large p small n" dilemma, common in datasets generated through bioinformatics techniques such as microarrays or next-generation sequencing, poses challenges [43]. For instance, while genomic variations can run into tens of thousands, sample sizes remain relatively smaller. Though our study demonstrates that using BLogic, by repeatedly applying SA and then combining the results using BMC, can address this, the sheer volume of explanatory variables still presents a hurdle. When the genuine key explanatory variables are substantially fewer than the total variables, directly applying SA might fail to find the true solution. Therefore, when confronted with datasets abundant in explanatory variables, various methods of integrating or transforming variable information become worth considering. Techniques such as feature selection [44,45], feature extraction [46,47], weighting variables [48,49], regularization [50], and split-and-merge [51] approaches can be integrated. Incorporating these into the our proposed BLogic algorithm may set the stage for a more streamlined inclusion of genuinely pivotal explanatory variables into the logic regression model, concluding our quest for enhanced predictability and interpretability in this field.

Funding: This work was partially supported by grants from the National Science and Technology Council of Taiwan (NSTC 112-2118-M-018-004).

Acknowledgments: The author thanks the National Center for High-performance Computing (NCHC) of the National Applied Research Laboratories (NARLabs) of Taiwan for providing computational and storage resources. Moreover, this study makes use of data generated by the UK10K Consortium, derived from samples from the Cambridge Severe Insulin Resistance Study Cohort and

the Twins UK Cohort. A full list of the investigators who contributed to the generation of the data is available at www.UK10K.org. Funding for UK10K was provided by the Wellcome Trust under award WT091310. Special acknowledgment goes to Yu-Xiang Liu and Chin-Yi Chao for their invaluable assistance in downloading, cleaning, and validating the data. Furthermore, the author extends sincere thanks to the diligent reviewers for their rigorous scrutiny and insightful feedback through multiple rounds of review. Their invaluable advice was crucial for refining the manuscript, resulting in a more coherent, robust, and valuable final paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Cordell, H.J. Detecting gene–gene interactions that underlie human diseases. *Nat. Rev. Genet.* **2009**, *10*, 392–404. [CrossRef] [PubMed]
2. Tekin, E.; Savage, V.M.; Yeh, P.J. Measuring higher-order drug interactions: A review of recent approaches. *Curr. Opin. Syst. Biol.* **2017**, *4*, 16–23. [CrossRef]
3. Kuhn, M.; Johnson, K.; Kuhn, M.; Johnson, K. Classification trees and rule-based models. In *Applied Predictive Modeling*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 369–413.
4. Apté, C.; Weiss, S. Data mining with decision trees and decision rules. *Future Gener. Comput. Syst.* **1997**, *13*, 197–210. [CrossRef]
5. Kocbek, S.; Kocbek, P.; Gosak, L.; Fijačko, N.; Štiglic, G. Extracting new temporal features to improve the interpretability of undiagnosed type 2 diabetes mellitus prediction models. *J. Pers. Med.* **2022**, *12*, 368. [CrossRef]
6. Bellavia, A.; Rotem, R.S.; Dickerson, A.S.; Hansen, J.; Gredal, O.; Weisskopf, M.G. The use of logic regression in epidemiologic studies to investigate multiple binary exposures: An example of occupation history and amyotrophic lateral sclerosis. *Epidemiol. Methods* **2020**, *9*, 20190032. [CrossRef] [PubMed]
7. Meijsen, J.J.; Rammos, A.; Campbell, A.; Hayward, C.; Porteous, D.J.; Deary, I.J.; Marioni, R.E.; Nicodemus, K.K. Using tree-based methods for detection of gene–gene interactions in the presence of a polygenic signal: Simulation study with application to educational attainment in the Generation Scotland Cohort Study. *Bioinformatics* **2019**, *35*, 181–188. [CrossRef]
8. Yoneoka, D.; Eguchi, A.; Nomura, S.; Kawashima, T.; Tanoue, Y.; Murakami, M.; Sakamoto, H.; Maruyama-Sakurai, K.; Gilmour, S.; Shi, S. Identification of optimum combinations of media channels for approaching COVID-19 vaccine unsure and unwilling groups in Japan. *Lancet Reg. Health–West. Pac.* **2022**, *18*, 100330. [CrossRef]
9. Rocco, C.M.; Hernandez-Perdomo, E.; Mun, J. Application of logic regression to assess the importance of interactions between components in a network. *Reliab. Eng. Syst. Saf.* **2021**, *205*, 107235. [CrossRef]
10. Li, T.; Sun, X.; Shu, X.; Wang, C.; Wang, Y.; Chen, G.; Xue, N. Robot grasping system and grasp stability prediction based on flexible tactile sensor array. *Machines* **2021**, *9*, 119. [CrossRef]
11. Lau, M.; Wigmann, C.; Kress, S.; Schikowski, T.; Schwender, H. Evaluation of tree-based statistical learning methods for constructing genetic risk scores. *BMC Bioinform.* **2022**, *23*, 1–30. [CrossRef]
12. Ruczinski, I. Logic Regression and Statistical Issues Related to the Protein Folding Problem. Ph.D. Thesis, University of Washington, Washington, DC, USA, 2001.
13. Otten, R.H.; van Ginneken, L.P. *The Annealing Algorithm*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 72.
14. Aarts, E.H. *Simulated Annealing: Theory and Applications*; Reidel: Dordrecht, The Netherlands, 1987.
15. Kooperberg, C.; Ruczinski, I.; LeBlanc, M.L.; Hsu, L. Sequence analysis using logic regression. *Genet. Epidemiol.* **2001**, *21*, S626–S631. [CrossRef]
16. Ruczinski, I.; Kooperberg, C.; LeBlanc, M. Logic regression. *J. Comput. Graph. Stat.* **2003**, *12*, 475–511. [CrossRef]
17. Fritsch, A.; Ickstadt, K. Comparing logic regression based methods for identifying SNP interactions. In Proceedings of the International Conference on Bioinformatics Research and Development, Berlin, Germany, 12–14 March 2007; pp. 90–103.
18. Wolf, B.J.; Hill, E.G.; Slate, E.H. Logic forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics* **2010**, *26*, 2183–2189. [CrossRef] [PubMed]
19. Schwender, H.; Ickstadt, K. Identification of SNP interactions using logic regression. *Biostatistics* **2008**, *9*, 187–198. [CrossRef]
20. Kooperberg, C.; Ruczinski, I. Identifying interacting SNPs using Monte Carlo logic regression. *Genet. Epidemiol.* **2005**, *28*, 157–170. [CrossRef]
21. Hubin, A.; Storvik, G.; Frommlet, F. A novel algorithmic approach to Bayesian logic regression (with discussion). *Bayesian Anal.* **2020**, *15*, 263–333. [CrossRef]
22. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]
23. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
24. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
25. Monteith, K.; Carroll, J.L.; Seppi, K.; Martinez, T. Turning Bayesian model averaging into Bayesian model combination. In Proceedings of the 2011 International Joint Conference on Neural networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2657–2663.

26. Domingos, P. Bayesian averaging of classifiers and the overfitting problem. In Proceedings of the International Conference on International Conference on Machine Learning, Stanford University, Stanford, CA, USA, 29 June–2 July 2000; pp. 223–230.
27. Minka, T.P. Bayesian Model Averaging Is Not Model Combination. 2002. Available online: https://tminka.github.io/papers/minka-bma-isnt-mc.pdf (accessed on 21 February 2021).
28. Kooperberg, C.; Ruczinski, I.; Kooperberg, M.C. Package 'LogicReg'. Comprehensive R Archive Network. 2015. Available online: http://cran.fhcrc.org/web/packages/LogicReg/LogicReg.pdf (accessed on 1 March 2021).
29. The UK10K Consortium. The UK10K project identifies rare variants in health and disease. *Nature* **2015**, *526*, 82–90. [CrossRef]
30. Li, H.; Handsaker, B.; Wysoker, A.; Fennell, T.; Ruan, J.; Homer, N.; Marth, G.; Abecasis, G.; Durbin, R. Genome Project Data Processing Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics* **2009**, *25*, 2078–2079. [CrossRef]
31. McKenna, A.; Hanna, M.; Banks, E.; Sivachenko, A.; Cibulskis, K.; Kernytsky, A.; Garimella, K.; Altshuler, D.; Gabriel, S.; Daly, M. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **2010**, *20*, 1297–1303. [CrossRef] [PubMed]
32. Joshi, S.R.; Pendyala, G.S.; Shah, P.; Pustake, B.; Mopagar, V.; Padmawar, N. Severe insulin resistance syndrome–A rare case report and review of literature. *Natl. J. Maxillofac. Surg.* **2021**, *12*, 100. [CrossRef] [PubMed]
33. Longo, N.; Wang, Y.; Smith, S.A.; Langley, S.D.; DiMeglio, L.A.; Giannella-Neto, D. Genotype–phenotype correlation in inherited severe insulin resistance. *Hum. Mol. Genet.* **2002**, *11*, 1465–1475. [CrossRef] [PubMed]
34. Sinnarajah, K.; Dayasiri, M.; Dissanayake, N.; Kudagammana, S.; Jayaweera, A. Rabson Mendenhall Syndrome caused by a novel missense mutation. *Int. J. Pediatr. Endocrinol.* **2016**, *2016*, 21. [CrossRef] [PubMed]
35. Kosztolanyi, G. Leprechaunism/Donohue syndrome/insulin receptor gene mutations: A syndrome delineation story from clinicopathological description to molecular understanding. *Eur. J. Pediatr.* **1997**, *156*, 253. [CrossRef]
36. Al-Beltagi, M.; Bediwy, A.S.; Saeed, N.K. Insulin-resistance in paediatric age: Its magnitude and implications. *World J. Diabetes* **2022**, *13*, 282. [CrossRef]
37. Tan, K.; Kimber, W.A.; Luan, J.a.; Soos, M.A.; Semple, R.K.; Wareham, N.J.; O'Rahilly, S.; Barroso, I. Analysis of genetic variation in Akt2/PKB-β in severe insulin resistance, lipodystrophy, type 2 diabetes, and related metabolic phenotypes. *Diabetes* **2007**, *56*, 714–719. [CrossRef]
38. An, P.; Freedman, B.I.; Hanis, C.L.; Chen, Y.-D.I.; Weder, A.B.; Schork, N.J.; Boerwinkle, E.; Province, M.A.; Hsiung, C.A.; Wu, X. Genome-wide linkage scans for fasting glucose, insulin, and insulin resistance in the National Heart, Lung, and Blood Institute Family Blood Pressure Program: Evidence of linkages to chromosome 7q36 and 19q13 from meta-analysis. *Diabetes* **2005**, *54*, 909–914. [CrossRef]
39. Van Tilburg, J.; Sandkuijl, L.; Strengman, E.; Van Someren, H.; Rigters-Aris, C.; Pearson, P.; Van Haeften, T.; Wijmenga, C. A genome-wide scan in type 2 diabetes mellitus provides independent replication of a susceptibility locus on 18p11 and suggests the existence of novel Loci on 2q12 and 19q13. *J. Clin. Endocrinol. Metab.* **2003**, *88*, 2223–2230. [CrossRef]
40. Dorajoo, R.; Liu, J.; Boehm, B.O. Genetics of type 2 diabetes and clinical utility. *Genes* **2015**, *6*, 372–384. [CrossRef] [PubMed]
41. Uffelmann, E.; Huang, Q.Q.; Munung, N.S.; De Vries, J.; Okada, Y.; Martin, A.R.; Martin, H.C.; Lappalainen, T.; Posthuma, D. Genome-wide association studies. *Nat. Rev. Methods Primers* **2021**, *1*, 59. [CrossRef]
42. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
43. Schena, M.; Shalon, D.; Davis, R.W.; Brown, P.O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **1995**, *270*, 467–470. [CrossRef] [PubMed]
44. Chen, J.; Aseltine, R.H.; Wang, F.; Chen, K. Tree-guided rare feature selection and logic aggregation with electronic health records data. *arXiv* **2022**, arXiv:2206.09107.
45. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature selection: A data perspective. *ACM Comput. Surv.* **2017**, *50*, 1–45. [CrossRef]
46. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the 2014 Science and Information conference, London, UK, 27–29 August 2014; pp. 372–378.
47. Guyon, I.; Gunn, S.; Nikravesh, M.; Zadeh, L.A. *Feature Extraction: Foundations and Applications*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 207.
48. Maudes, J.; Rodríguez, J.J.; García-Osorio, C.; García-Pedrajas, N. Random feature weights for decision tree ensemble construction. *Inf. Fusion* **2012**, *13*, 20–30. [CrossRef]
49. Chen, Y.-C. *An Ensemble Logic Regression Approach for Detecting Important Genes and Interactions*; National Changhua University of Education: Changhua, Taiwan, 2023.
50. Lim, M.; Hastie, T. Learning interactions via hierarchical group-lasso regularization. *J. Comput. Graph. Stat.* **2015**, *24*, 627–654. [CrossRef]
51. Huang, W.-H.; Wei, Y.-C. A split-and-merge deep learning approach for phenotype prediction. *Front. Biosci. Landmark* **2022**, *27*, 78. [CrossRef] [PubMed]