Human Genetics (2022) 141:1515–1528
https://doi.org/10.1007/s00439-021-02402-z

REVIEW

What makes a good prediction? Feature importance and beginning to open the black box of machine learning in genetics

Anthony M. Musolf[1] · Emily R. Holzinger[2] · James D. Malley[1] · Joan E. Bailey-Wilson[1]

Received: 13 May 2021 / Accepted: 8 November 2021 / Published online: 4 December 2021

Abstract

Genetic data have become increasingly complex within the past decade, leading researchers to pursue increasingly complex questions, such as those involving epistatic interactions and protein prediction. Traditional methods are ill-suited to answer these questions, but machine learning (ML) techniques offer an alternative solution. ML algorithms are commonly used in genetics to predict or classify subjects, but some methods evaluate which features (variables) are responsible for creating a good prediction; this is called feature importance. This is critical in genetics, as researchers are often interested in which features (e.g., SNP genotype or environmental exposure) are responsible for a good prediction. This allows for the deeper analysis beyond simple prediction, including the determination of risk factors associated with a given phenotype. Feature importance further permits the researcher to peer inside the black box of many ML algorithms to see how they work and which features are critical in informing a good prediction. This review focuses on ML methods that provide feature importance metrics for the analysis of genetic data. Five major categories of ML algorithms: k nearest neighbors, artificial neural networks, deep learning, support vector machines, and random forests are described. The review ends with a discussion of how to choose the best machine for a data set. This review will be particularly useful for genetic researchers looking to use ML methods to answer questions beyond basic prediction and classification.

Introduction

In the past decade, genetic data have become more and more complex. Researchers now have easy access to whole-genome sequence (WGS), whole-exome sequence (WES), RNA sequencing (RNA-seq), and other forms of gene expression and proteomics data. As the data size and complexity has increased, the questions geneticists have begun to answer have increased at a commensurate rate. For example, researchers now want to know whether disease risk is caused by epistatic (gene–gene) interactions between germline genetic variants or what effect a single missense variant in the DNA might have on the structure, and consequently the function, of a protein. With traditional methods often struggling to provide sufficient answers to these questions, geneticists have begun to frequently turn to machine learning (ML) techniques (Moore et al. 2010; Moore and Williams 2009).

ML methods can be broadly defined as the algorithms (machines) that can learn from data to make predictions or identify patterns too complex for humans to detect. ML uses a sample set of data, called a training set, to initially program or construct the prediction model. Further input into the machine will then give an output based on how the machine was trained. ML is considered a branch of artificial intelligence (AI).

There are multiple types of ML techniques, many of which have been adapted to genetic data. The vast majority of ML algorithms in genetic analysis are used in prediction, essentially using some input data to predict an outcome or the value of a trait for a particular subject. In genetic data, this could be using SNP genotypes to predict whether a subject is a case or control or using gene expression data to predict whether a tumor is likely to metastasize.

Good predictions are indeed critical in genetics. However, identifying which variables (in ML parlance called features) are most responsible for informing these good predictions is

✉ Joan E. Bailey-Wilson
jebw@mail.nih.gov

[1] Statistical Genetics Section, Computational and Statistical Genomics Branch, National Human Genome Research Institute, National Institutes of Health, 333 Cassell Drive Suite 1200, Baltimore, MD 21224, USA

[2] Target Sciences, Informatics and Predictive Sciences, Bristol Myers Squibb, Cambridge, MA, USA

also critical. The difference between prediction, estimation, and attribution has been addressed by many authors, and is well described by, for example, Efron (2020), Malley et al. (2011), and Hastie et al. (2009). ML algorithms, especially the more sophisticated ones, seem like black boxes. You put in some data and get out a prediction. However, in genetics, we not only want to get a good prediction—we want to know which variables, such as age, sex, environmental exposures, and genetic variant genotypes, are important at predicting disease or predicting which genes are differentially expressed in metastatic versus benign tumors.

This is called feature importance in ML—knowing which features are the best predictors. Feature importance (which is also called feature detection, feature attribution, or model interpretability and is related to the statistical ideas of estimation and attribution) will output a particular score or metric, permitting ranking of features from largest to smallest contribution to the machine's prediction. They are often obtained by systematically permuting features, to determine which feature causes the largest change in predictive power. This will produce an importance score for each feature, allowing for feature ranking. For example, this would allow researchers to identify which genetic variants might be associated with disease and then earmark them for functional studies. It would also allow a protein prediction program to identify which set of input data were most important at creating a good prediction. Thus, it is important to keep in mind while reading this manuscript that different feature importance metrics measure different things and are thus not comparable; indeed, many importance measures are not interpretable in general.

This review sets out to outline some of the different software programs that provide feature importance metrics using genetic data; it is not meant to be an exhaustive review of the subject. For the scope of this paper, genetic data will be defined as the analysis of DNA or RNA sequence data. Much of this paper will concern two major topics. The first is determining which genetic variants are associated with a given disease, specifically which variants increase the risk of disease. Here, the predictor variable will be variant genotypes and the dependent variable will be disease status, which can be binary or quantitative. Thousands of variant genotypes (millions with whole-genome sequence data) are analyzed and typically subjects number in the hundreds or thousands, dependent on the prevalence of the disease being studied. An important offshoot of this topic is finding epistatic interactions between variants, i.e., which variants interact to form an increased disease risk. The second topic we will address is variant pathogenicity. Here, changes in the DNA are extrapolated to protein structures, where one will attempt to predict the damage to the protein structure based on the amino acid substitution caused by the DNA mutation. Here, the independent variable is the variant genotype, while the dependent variables are the amino acid changes and its subsequent effect on the protein. However, we also touch on other uses of machine learning such as approaches that involve RNA sequence data for gene expression analyses.

Feature importance will be discussed across five of the most popular machines—*k* nearest neighbors, artificial neural networks, deep learning, support vector machines, and random forest—before discussing some approaches for choosing the best machine for a particular data set as well as tuning machine parameters. A tabular form of the software discussed is provided in Table 1.

Methods

K nearest neighbors

K nearest neighbors (*k*-NN) is one of the simplest and oldest ML algorithms. However, it can still be as effective as some of the newer, more complex machines (Malley et al. 2011). The basic assumption that underlies the *k*-NN approach is that the classification of a subject into a group (such as case or control in a genetic study) should depend primarily on the other subjects closest to it, its "neighborhood". While the term neighborhood sounds somewhat nebulous, it is determined by a distance metric between subjects, with the Euclidean distance being a popular choice (Fig. 1). In practice, this works by plotting all subjects in space based on a set of features (e.g., SNP genotypes). For each subject, the machine looks at a predetermined number (termed k) of neighbors with the shortest distance from the subject and a majority vote of these neighbors then determines the subject's classification. *k*-NN can also be used as a regression analysis for quantitative traits (Devroye et al. 1994).

k-NN has been shown to be effective despite its simplicity and has been effectively used in genetic studies (Malley et al. 2011). In genetics, other distances besides Euclidean are often used, such as Mahalanobis distance, which allows for effects like correlation (critical for linkage disequilibrium between variants) (Abo Alchamlat and Farnir 2017) and Hamming distance, which is useful for strings of data (like DNA sequence). For genetic data, identity-by-state (IBS) and identity-by-descent (IBD) distance can also be used; both NetView (Neuditschko et al. 2012) and FastSMC (Nait Saada et al. 2020) use IBD values to determine fine-scale population structures.

Despite its effectiveness, *k*-NN has some major drawbacks. First, the machine requires certain input parameters to run, specifically the choice of k and the type of distance measurement (Abu Alfeilat et al. 2019). Further *k*-NN is computationally intensive, as the distance between all subjects needs to be calculated (Malley et al. 2011). To combat this, filter methods are often applied to decrease the total

Table 1 List of machine learning software

Software name	Machine type	Software type	Application	Data type	Website
SURF	*k*-NN	Part of open-source package	Epistatic interactions	SNP	http://www.epistasis.org
STIR	*k*-NN	Standalone Program	Epistatic interactions	SNP	http://insilico.utulsa.edu/software/STIR
ReliefSeq	*k*-NN	Standalone Program	Epistatic interactions	RNA-seq	http://insilico.utulsa.edu/ReliefSeq.php
KNN-MDR	*k*-NN	Standalone program	Epistatic interactions	SNP	n/a
GANN	ANN	Standalone program	Gene-based expression association	RNA-seq	n/a
ANNI	ANN	Standalone program	Epistatic interactions	SNP	n/a
ATHENA	ANN	Standalone program	Epistatic interactions	SNP	https://ritchielab.org/software/athena-downloads
Basset	Deep learning	Standalone program	Noncoding annotation	DNA-seq	https://github.com/davek44/Basset
DeepSEA	Deep learning	Standalone program	Noncoding annotation	DNA-seq	http://deepsea.princeton.edu/
DeepWAS	Deep learning	Standalone program	GWAS/annotation integration	GWAS	https://github.com/cellmapslab/DeepWAS
DFIM	Deep learning	Standalone program	Epistatic interactions	DNA-seq	https://github.com/kundajelab/dfim
PrimateAI	Deep learning	Standalone program	Variant pathogenicity	DNA-seq	https://basespace.illumina.com
CADD	SVM	Standalone program	Variant pathogenicity	DNA-seq	https://cadd.gs.washington.edu
MSIpred	SVM	Python package	Microsatellite instability prediction	WES	https://github.com/bioinfolabmu/MSIpred
REVEL	RF	Standalone program	Variant pathogenicity	DNA-seq	https://sites.google.com/site/revelgenomics/
Random jungle	RF	R package	GWAS	SNP	https://r-forge.r-project.org/R/?group_id=741
Ranger	RF	R package	GWAS	SNP	https://cran.r-project.org/web/packages/ranger/index.html
Open target genetics	RF	Standalone program	SNP/gene prioritization	GWAS Results	https://genetics.opentargets.org
Permuted RF	RF	Standalone program	Epistatic interactions	SNP	n/a
RF fishing	RF	Standalone program	Epistatic interactions	SNP	n/a
SWSFS	RF	Standalone program	Epistatic interactions	SNP	n/a
r2VIM	RF	Standalone program	Epistatic interactions	SNP	https://research.nhgri.nih.gov/software/r2VIM/
Boruta	RF	R package	Epistatic interactions	SNP	https://cran.r-project.org/web/packages/Boruta/index.html
Vita	RF	R package	Epistatic interactions	SNP	https://cran.r-project.org/web/packages/vita/index.html

A list of the software referenced in this review. The columns represent the software names, the type of machine used in the software, the type of software (i.e., whether the software is a stand-alone program or a package), the application for the software, the type of data the software analyzes (note that programs that use SNP data can also use DNA-seq), and the link to download the software (if available)

number of features used in *k*-NN analyses. This could mean using other ML methods to reduce feature input or using annotation, for instance biological function, to whittle down the total number of features.

Feature importance can be determined in *k*-NN. Relief algorithms have overcome the dimensionality issues using wrappers and filters to decrease computation time (Greene et al. 2009; Moore et al. 2006), allowing for assessment of feature importance. Filters rank variables based on a metric that estimates the association with the outcome, independently of the ML method. In contrast, wrappers use the ML algorithm on subsets of the variables to select an optimal set. This has been particularly useful in the study of epistasis in GWAS or sequence data. Spatially Uniform Relief (SURF)

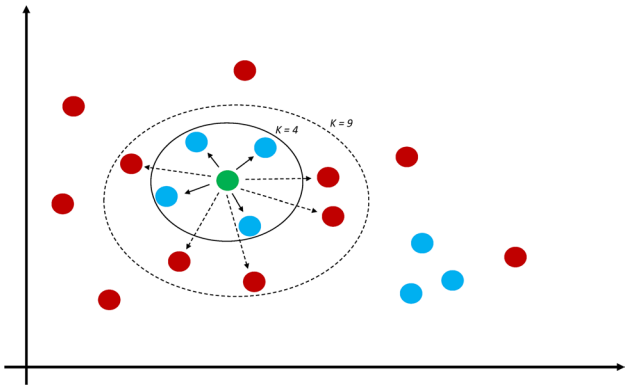

Fig. 1 *k*-nearest neighbors. A diagram showing an example of the k-nearest neighbor machine. Subjects are plotted based on feature values, and an individual's classification is determined by a majority vote in the subject's neighborhood (*k*). The choosing of *k* is crucial to classification. For instance, if we wished to classify the green individual based on $k=4$, the individual would be classified as blue. If we extended this to $k=9$, the individual would be classified as red

assigns each feature a score and has been found to effectively identify pairwise interacting SNPs, even when those SNPs do not have main effects (Greene et al. 2009). While SURF cannot identify the nature of the interaction between the SNPs, it can filter the features based on a Relief score threshold to a number feasible for traditional combinatorial approaches. This list of important features can be used as input to other approaches that can model interactions. A more recent update of the SURF approach, called statistical inference relief (STIR), was developed to calculate feature (attribute) scores without the need for permutation (Le et al. 2019) and was successfully applied to RNA-Seq data. Other genetic uses of Relief algorithms include ReliefSeq, which has been adapted as a gene-based test to find both main effects and gene–gene interactions in mRNA-seq expression data (McKinney et al. 2013).

The KNN-MDR approach (Abo Alchamlat and Farnir 2017) combines *k*-NN with multifactor dimensionality reduction (MDR). This approach substitutes the majority vote of MDR amongst individuals sharing the same genotypes with a majority vote of the *k*-NN of a subject. *P* values are produced for each SNP pair, and the method has > 70% power to detect simulated disease SNPs under two-way and three-way interaction scenarios in simulations with 500 cases and 500 controls.

Random forest

The next machine that will be discussed in this review is random forest (RF). RF is built upon the concept of the classification and regression tree (CART), which takes a group of heterogeneous data and repeatedly splits the original data set into more homogeneous groups (termed nodes) based on features (Breiman 2001; Malley et al. 2011) (Fig. 2a). Deciding which feature to initiate the split at a particular node is often determined by its increase in purity in the resulting nodes. Purity is the measure of class homogeneity at a particular node; for instance, the percentage of cases vs. controls. The higher the proportion of one class, the purer the node. There are multiple measures of purity, including

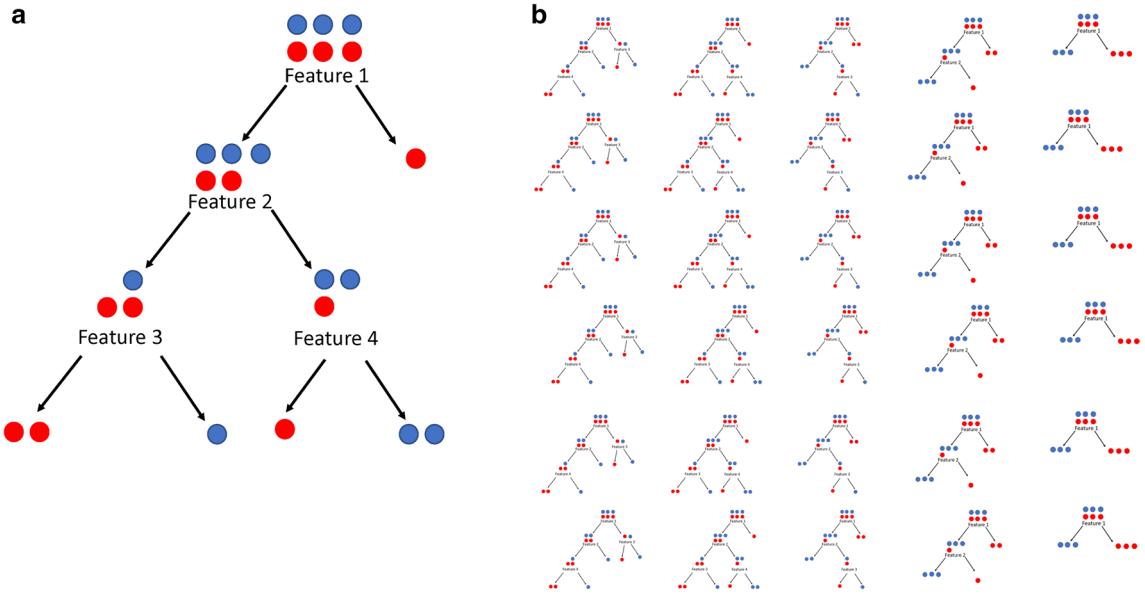

Fig. 2 Classification and Regression Trees (CART) and Random Forest. **a** Diagram showing a single CART. CARTs take a heterogeneous group of data and repeatedly split on feature values to create more homogeneous groups. **b** Diagram showing a random forest. A random forest is a collection of CARTs, each running on a slightly different subset of the same data set

the popular Gini index (Malley et al. 2011). Splitting ceases when a predetermined purity threshold is reached; nodes that are not split further are called terminal nodes. The subjects in the terminal nodes are tallied and a simple majority then determines node classification (Malley et al. 2011). RF is just a collection of CARTs (hence the name "forest") that are built upon bootstrap samples of the original dataset. Classification is determined by a majority vote across the trees of the forest (Breiman 2001) (Fig. 2b). A genetics example using RF would be the use of classification trees to perform repeated splitting of case/control data by SNP genotypes to determine which alleles affect case classification. RF can be run as regression trees on continuous data, as well.

The inherently model-free approach has made RF a popular machine in genetic analysis. RF has been used in the prediction of variant pathogenicity (Ioannidis et al. 2016), association studies (Szymczak et al. 2009), RNA-seq and expression data (Guo et al. 2020), next-generation sequencing calling quality control (Li et al. 2019), and DNA methylation (Wilhelm 2014). RF is often used for purely predictive purposes; however, feature importance (often called variable importance in RF parlance) is relatively trivial to calculate in RF. Thus, there are more examples of methods to determine feature importance in RF than for other machines.

There are several different ways that feature importance can be calculated in RF. One method is the Gini importance. As mentioned above, at each node, features are assigned a Gini index (a measure of node purity) to determine which feature to perform the split. These Gini indices for features can be averaged across all nodes and trees to determine the Gini importance of the feature in the analysis. Gini importance has been shown to have biases, including toward higher frequency features (Nicodemus 2011), which would include bias toward more common SNPs in genetic studies (Boulesteix et al. 2012). A recent approach by Nembrini et al. (2018) has sought to remove these biases and has been shown to be powerful in genetic analysis.

Another major method of calculating feature importance in RF is permutation, where features are systematically permuted and their effect on prediction is observed (Malley et al. 2011). Importance scores can then be created for each feature, allowing for the ability to rank features by importance (Szymczak et al. 2016). The permutation approach lacks the biases of the Gini importance, but takes significantly longer to compute, as each feature needs to be permuted individually across all the trees in the forest while keeping the other features constant. It should be noted that permutation-based importance has biases, as well, including increased false positives due to unimportant features correlated with important features. Several conditional corrections have been proposed, including the permutation importance (PIMP) approach (Altmann et al. 2010) and the conditional permutation approach (Strobl et al. 2008). However, in genetic association studies, detection of association to groups of variants that are in strong LD with each other are not considered serious false positives, because they direct the researcher to the region of the genome that may harbor a causal variant. Random jungle (RJ) is a software package that can rapidly analyze GWAS data using RF (Schwarz et al. 2010). RJ can calculate feature importance based on both Gini importance and permutation-derived importance. It also includes a backwards elimination method, in which RFs are iteratively fitted, with features with low importance scores being removed at each step. RJ was used to identify new genes associated with Crohn's disease (Schwarz et al. 2010). The initial RJ software has been subsumed and improved by the R package ranger, which is even more efficient at analyzing high-dimensional data (Wright and Ziegler 2017).

RF has become popular for predicting variant pathogenicity. One such example is the rare exome variant ensemble learner (REVEL). REVEL is a RF-based method that is used to predict the pathogenicity of rare missense variants (Ioannidis et al. 2016). REVEL was trained with recently discovered pathogenic and neutral variants and takes as input functional scores from well-known prediction programs like SIFT (Ng and Henikoff 2003) and FATHMM (Shihab et al. 2013) and conservation scores like GERP++ and SiPhy. Not only did REVEL have better overall performance than other methods, but it was able to identify which features were most important in REVEL's pathogenicity predictions (Ioannidis et al. 2016). REVEL found that the functional scores from FATHMM (Shihab et al. 2013) and VEST (Carter et al. 2013) were the most important features in its pathogenicity predictions, and that functional scores in general were more important than conservation (Ioannidis et al. 2016). These feature importance scores give us valuable insight into what is going on within REVEL's black box, as well as let us know that FATHMM and VEST may be more effective than other functional annotation programs of interest and that, perhaps, we should put greater weight on functional scores compared to conservation scores.

In a similar manner, feature importance allows us to peek inside the black box of Open Target Genetics' prioritization machines. These prioritization machines are based on a gradient boosting classifier (not strictly a RF but uses CARTs like RF) (Ghoussaini et al. 2021). Locus-to-gene (L2G) is one such machine that incorporates different features like distance to gene, expression data, and chromatin interactions to prioritization causal genes from a number of SNPs from a significant locus (Ghoussaini et al. 2021). This is especially useful when looking at many noncoding SNPs, or SNPs along a linked or associated haplotype. While an L2G score is given for each gene, feature importance metrics

show that distance to gene is more important than other factors like gene expression.

RF and other ML methods are used in phenotype definition, as well. While most genetic analyses focus primarily on genotypes, the characterization of phenotypes can be just as critical. ML approaches can be used to find disease subgroups that might not be identified through traditional analyses. This sort of clustering analysis is an example of unsupervised learning, where machines (such as RF) look for hidden patterns in the dataset (Shi and Horvath 2006). This is opposed to supervised ML, where machines are looking to accurately classify or predict (the subject of the majority of this manuscript).

Crucial to this effort is the analysis of electronic health records (EHRs), which contain copious amounts of diverse data on patients that can be mined to elucidate homogeneous subgroups from heterogeneous traits (Basile and Ritchie 2018). These analyses can be used to identify unique disease subsets through prediction or identify novel sets of features to better classify affected subjects through feature importance. For instance, Teixeira et al. used numerous ML techniques to analyze EHRs to identify individuals with hypertension, with RF being the most effective (Teixeira et al. 2017). Looking at feature importance determined that blood pressure measurements, which is traditionally used to diagnose clinical hypertension, was the worst-performing feature at predicting hypertension. Other EHR information, such as vitals and medications, was much more effective (Teixeira et al. 2017).

The detection of epistatic interactions between variants is another popular area of active research in RF feature importance, even in high-dimensional data (Lunetta et al. 2004; Winham et al. 2013). RF is well suited to identify interactions under the theory that if variants are indeed interacting, then it is likely that once one variant is chosen as a splitting criterion, the interacting variant may shortly follow. This will rank both features as significant and also provides a mechanism to identify higher order interactions (Holzinger et al. 2016). While there are multiple ways to determine RF feature importance in epistasis, Orlenko and Moore determined that permutation-derived importance metrics are more precise at identifying interactions (Orlenko and Moore 2021).

Various flavors of RF have been developed to detect epistatic interactions amongst variants. For instance, permuted random forest (pRF) identifies interacting SNP pairs by systematically permuting interactions between a pair of SNPs and determining which SNP pairs cause the greatest reduction in prediction power (Li et al. 2016). Random forest fishing (RFF) is an iterative approach that has been shown to identify important variants even when no main effects are present on the variants (Yang and Charles Gu 2014).

Sliding window sequential forward feature selection (SWSFS) uses SNP genotypes as categorical features and uses a sliding window approach to select a small number of candidate SNPs that minimized classification error using Gini importance, as opposed to permutation-derived importance (Jiang et al. 2009). SWSFS was used to test up to three-way interactions and was successfully used to identify SNPs associated with age-related macular degeneration.

Recurrent relative variable importance measure (r2VIM) adds the principle of recurrency to RF to identify epistatic interactions in SNP genotype data (Szymczak et al. 2016). In a single run of RF, false positives may have higher importance scores than true predictors simply by chance. Recurrency eliminates this problem by running multiple independent RF analyses on the same data set, using a different starting seed. Permutation-based importance scores are then calculated for each feature for each of the analyses, which are termed relative importance scores. The median of these relative importance scores is then taken to represent the true importance score. This serves to reduce false positives while keeping true predictors with large importance scores. r2VIM was shown to control false-positive rates and identify main effect SNPs (Szymczak et al. 2016). It has also been shown to identify epistatically interacting SNPs as important, even when these SNPs have no main effects on the trait (Holzinger et al. 2015).

One might wonder with multiple ways to calculate feature importance in RF, is one method more effective than the other? Degenhardt et al. undertook a comparison of feature selection methods, including standard permutation-derived approaches and the recurrent approach of r2VIM (Degenhardt et al. 2019). Other evaluated approaches included that of Boruta, which calculates importance scores by creating shadow features (shadow variables) by doubling each feature and permuting it. The importance scores of the real features are then compared to that the shadow features (Kursa and Rudnicki 2010). A fourth method was that of Vita, which involves dividing the overall data into two equal, independent subsets and estimating variable importance using the other set; an importance score called the hold-out importance is calculated (Janitza et al. 2018). After comparison, Degenhardt et al. concluded that Boruta and Vita were the most powerful approaches in simulation studies using high-dimensional data, noting that Vita is significantly faster (Degenhardt et al. 2019).

Artificial neural networks

Artificial neural networks (ANNs), also known as neural networks (NNs), are a more complex type of machine. They consist of multiple small models (nodes) linked together, feeding the output of one model into the input of another. In this way, they loosely resemble the neurons and synapses

of the brain as information is rapidly transmitted from one model to another. ANNs take input data that are given to an initial set of small models (which can just be simple models like linear/logistic regression). The output from the first model is then transmitted to a second set of models as a weighted sum. This process is then repeated as multiple small models are combined into distinct weighted sums until a final output is reached. The intermediate sums, which are not reported, are referred to as the hidden layers (Malley et al. 2011). Complex mechanisms such as backpropagation, where information is passed in the reverse direction to better fit the network, and gradient descent, an optimization algorithm used to minimize predictive loss, are used in this process. Traditional ANNs rarely have more than one or two hidden layers (Fig. 3). Furthermore, due to the often sparse signals in genetic data, fully connected ANNs (meaning each "neuron" in one layer feeds its data into every other "neuron" in the succeeding layer) are not always used. Instead, convolutional layers are used, where "neurons" are only connected some of the "neurons" in the succeeding layer.

Due to their model-free assumptions and their ability to interpret a wide range of data types, ANNs have been frequently used in genetics. They are especially popular prediction tools, whether it be for protein structure and folding (Cai et al. 2003), variant genotype calling in microarrays and sequencers (Poplin et al. 2018), or the question of whether tumors will metastasize (Wang and Yu 2020).

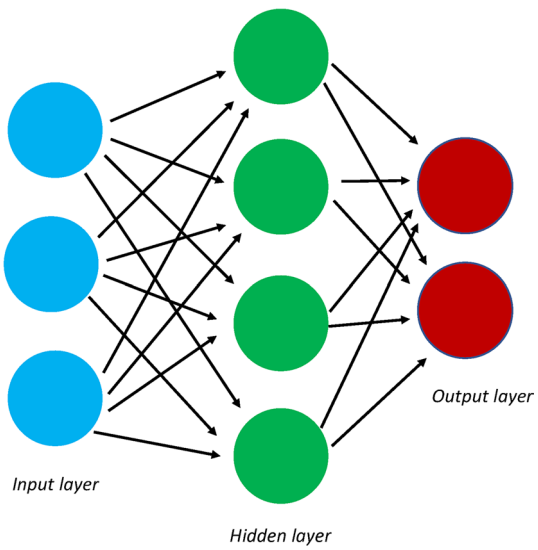

Fig. 3 Artificial neural networks. A schematic of an artificial neural network. Data are analyzed by different models, the results of which are passed onto a new set of models. In this example, data are first analyzed in the input layer (blue). The results are then passed onto an intermediate layer, called a hidden layer (green). Finally, the results of the hidden layer are passed onto and analyzed by the models of the output layer (red)

ANNs can be used for feature importance, as well, though they are more commonly used for prediction and are more of a black box than other methods. Olden and Jackson proposed a randomization approach for ANNs that allows users to quantitatively assess both the individual and interactive effects of the input variables in the network prediction process, as well as evaluate the overall contributions of the variables to the prediction, which they demonstrated effectively using ecological examples (Olden and Jackson 2002). Initially, in genetics, ANNs were used as an alternative to traditional single locus and multilocus association methods, meaning that ANNs would evaluate the potential association of a single marker on disease status. Curtis found their power comparable or better to other traditional methods like haplotype analysis (Curtis 2007), but there has not been much movement in this aspect for over a decade, likely due to the effectiveness of traditional methods at finding main effects in association studies.

However, this does not mean that feature importance using ANNs in genetics has remained stagnant. ANNs have been particularly useful in gene expression and gene–gene interaction studies. Tong and Schierz (2011) developed a hybrid ANN algorithm for gene expression data that specifically detects genes that are good predictors, called genetic analysis neural network (GANN). The ANN algorithm is combined with a genetic programming (GP) algorithm. GP algorithms are inspired by evolution; they generate a population of solutions which proceed to mutate and recombine over a preset number of generations. The best solutions are determined via a fitness metric. GANN emphasizes importance of features, and uses the ANN part of the algorithm to determine the fitness function of the GA. Results using expression data found that GANN was able identify genes that had been identified as significant using the traditional methods as well as novel genes that were biologically relevant to the trait being studied (Tong and Schierz 2011). The significant genes from the initial GANN study were later used to develop the artificial neural network inference (ANNI) approach to identify epistatic interactions. This procedure used ANNI to explore all potential influences of genes amongst themselves (Tong et al. 2014). A matrix of interactions that can be ranked by value is the output.

The Analysis Tool for Heritability and Environmental Network Associations (ATHENA) uses an ANN algorithm to perform a suite of analyses using multiple types of input data, including microarray, sequence, and expression data (Holzinger et al. 2014). ATHENA is designed to test pairwise interactions between variants and uses a modified ANN called a grammatical evolution neural network (GENN). GENNs transcribe input data, such as SNP genotypes, into an internal grammar to increase efficiency (Turner et al. 2010). They proceed in a manner similar to GP, evolving the heterogeneous mix of weights and inputs that undergo

mating crossovers and recombinations that test two-SNP models (Holzinger et al. 2014). Fitness is recorded for each model and models with the highest fitness are selected for crossover and reproduction; this is done for preset number of generations. ATHENA uses additional biological information to create its two-SNP interaction models, including pathway information from KEGG and functional information from Gene Ontology (Holzinger et al. 2014). ATHENA's output includes all features from the best model as well as their cross-validation scores; letting users observe which features have informed the best model.

Deep learning

A specialized type of ANN that has gained popularity in recent years is called deep learning. Deep learning relies upon deep neural networks, which follow the same principle as ANNs. Initial models compute some result from input and the output from the initial model is transmitted to another model. The process is repeated, producing more complex outputs along the way. However, the deep neural networks that underpin deep learning contain many hidden layers (as opposed to just one or two in ANNs) (Fig. 4). Recall the hidden layers are the intermediate models between the first input model and the final output model that is reported. Deep learning also can contain far more complex architecture than simple multilayered ANNs, including convolutional or recurrent layers. Deep learning has exploded in a variety of fields within recent years due to its ability to handle extremely complex, heterogeneous data (including image data), its model free assumptions, and its relative ease of use for non-experts.

In genetics, deep learning has been applied to DNA sequence data to predict the effect of noncoding variants (Zhou et al. 2018; Zhou and Troyanskaya 2015), pathogenicity of exonic variants (Sundaram et al. 2018), and the identification of regulatory motifs (Kelley et al. 2018) and transcription factor-binding sites (Wang et al. 2018). Given their proliferation in both biology and genetics in particular in the past 5 years, a full review of deep learning in genetics is well beyond the scope of this paper, but the interested reader is directed to an excellent review by Eraslan et al. (2019).

The vast majority of deep learning applications in genetics is concerned with prediction. It is possible to estimate feature importance, but Eraslan et al. noted that benchmarking and thresholding of importance scores in genomic data have not been well tested and should be compared to known simulated data (Eraslan et al. 2019). Feature importance in deep learning is calculated either by perturbation (changing a value and recording the subsequent change in prediction) or by backpropagation through the network (Eraslan et al. 2019).

Many of the pathogenicity and regulatory prediction programs have some sort of importance metric to determine which features affect their models. For instance, the regulatory sequence prediction program Basset (Kelley et al. 2016) uses perturbation of stretches of DNA to determine which sequence motifs are important to predicting regulatory sequences. DeepSEA (Zhou and Troyanskaya 2015), which predicts the effect of noncoding variants, uses a similar perturbation approach by changing single nucleotides of sequence data. DeepSEA then provides a functional significance score for noncoding variants based on chromatin effect predictions and evolutionary conservation. Regulatory predictions from DeepSEA have recently been integrated into a new program called DeepWAS (Arloth et al. 2020). While most functional annotation of GWAS is performed post hoc, DeepWAS combines the association analysis and functional

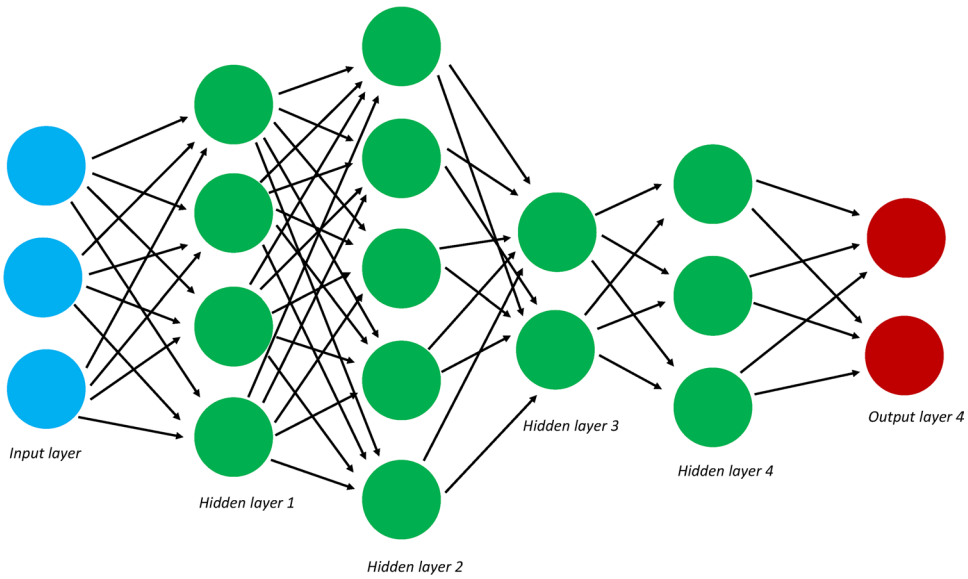

Fig. 4 Deep learning. A schematic of a deep learning machine. Deep learning is a specialized version of artificial neural networks that contain many additional hidden layers

annotation from DeepSEA in a single step to identify disease-associated loci that are likely to be regulatory variants. DeepWAS was successfully tested on real data, including multiple sclerosis and height.

Deep Feature Interaction Maps (DFIM) (Greenside et al. 2018) is a deep learning machine created to identify epistatic interactions between sequences of the genome. DFIM computes a novel Feature Interaction Score (FIS) between a target sequence and source sequence by systematically perturbing nucleotides in the source feature. A computationally efficient backpropagation is then used to calculate FIS between all pairs of nucleotides or regulatory motifs in a DNA sequence, determining which changes have the greatest effect. DFIM was used to identify synergistic interactions GATA1 and TAL1 motifs as well as other regulatory genetic variants (Greenside et al. 2018).

PrimateAI, a deep neural network that is trained on using the primary amino acid sequences, uses variants from non-human primates like chimpanzees, gorillas, and orangutans to better predict classification of pathogenic variants in humans (Sundaram et al. 2018). Using a deep neural network built to extract features from just the primary amino acid sequence of the variant of interest and flanking variants, PrimateAI was able to predict variant pathogenicity at a higher percentage than other prediction programs and identify novel candidate genes for intellectual disability. While the goal of PrimateAI was classification (they successfully estimated pathogenicity on over 70 million variants), the algorithm does look at effects of features on its neural network. For instance, it was found that each of six non-human primate genomes increased prediction accuracy while adding non-primate mammalian genomes (e.g., pig or cow) decreased accuracy (Sundaram et al. 2018). This led the authors to note that additional sequencing of non-human primates will increase pathogenicity prediction in humans. This is a nice example of how opening up the black box of these complex deep neural networks and looking at the actual effect of features on prediction leads to valuable feedback about how to improve the overall machine.

Support vector machines

Support vector machines (SVMs) are a large class of ML algorithms that have become popular in part because of their mathematical elegance and their ability to handle large numbers of features (Malley et al. 2011). SVMs work this way—within a set of data to be classified, there exists a decision boundary that can be drawn through the data to enable classification. SVMs orient this boundary, which is called the hyperplane, so that it is as far as possible from the two closest points of each class (Huang et al. 2018) (Fig. 5). A kernel function is used in higher dimensional models to calculate the hyperplane more efficiently. Essentially, the

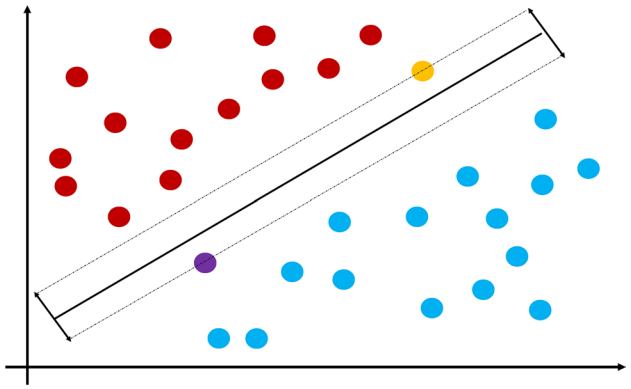

Fig. 5 Support vector machines. A diagram showing an example of a support vector machine. Subjects are plotted based on feature values, and a special boundary called the hyperplane is formed to classify individuals. The hyperplane is oriented as far as possible from the two closest individuals in each class (in this example, the orange and purple individuals)

kernel allows for datapoints in multiple dimensions to be treated as linear data, thus easily computing the distance between datapoints. Without the kernel function, this would be much more difficult. There are several different types of kernel functions, which are expounded on in the excellent text by Schölkopf et al. (2003). Popular kernels for SVMs include linear kernels, radial basis function (RBF) kernels for non-linear data, as well as Gaussian and polynomial kernels. One particular kernel of note for genetic data is the string kernel, which takes as input (long or short) sequences of text (called strings in programming). This is very useful in DNA-seq analysis where long stretches of genotypes can be compared. SVMs can classify continuous data, as well; in these cases, the SVM is usually referred to as support vector regression (SVR).

As they are powerful classifiers, SVMs are used for prediction in a variety of genetic scenarios. For instance, SVMs have been trained to use RNA-seq data to identify patients with thyroid cancer (Shen et al. 2020), proteomic data to identify different breast cancer subtypes (Tyanova et al. 2016), sequence data to detect somatic tumor mutations (Mao et al. 2021), and cell-free DNA to diagnose cancer (Liu et al. 2021).

One of the best known SVMs in genetics studies is Combined Annotation-Dependent Depletion (CADD) (Kircher et al. 2014; Rentzsch et al. 2019), a popular annotation program for identifying variant pathogenicity. CADD draws from a variety of different features including allelic diversity, annotations of functionality, pathogenicity, disease severity, and experimentally measured regulatory effects to create a quantitative CADD score, which can be used to prioritize deleterious exonic variants.

Though they are primarily used as predictors, SVMs offer feature importance metrics as well. One notable SVM is the

support vector recursive feature elimination (SVM-RFE) (Guyon et al. 2002). SVM-RFE first runs on all features and applies an importance score to each feature that is based on how well each feature classifies the training data. Lowest scores are iteratively removed, and the algorithm stops when all features are important. Guyon et al. first successfully used this method to select genes for cancer prediction (Guyon et al. 2002).

Hu et al. (2016) used SVM-RFE in an algorithm containing SVM in combination with random forest (RF) to identify genes responsible for cell differentiation. Using single-cell RNA-seq data from neocortical cells and neural progenitor cells, the SVM-RFE/RF classifiers were able to identify 38 genes that best predicted the differentiation of neocortical cells from the neural progenitor cells. Similarly, Xu et al. (2017) were able to use SVMs trained on colon cancer microarray expression data to identify 15 genes as predictors of recurrence risk and prognosis in colon cancer patients.

SVMs have been also used for SNP selection, as well. De Oliveira et al. (2014) used support vector regression to identify SNPs that were associated with a simulated phenotype. The method was able to identify at least some of the causal SNPs, even in the case of polygenic and epistatic effects. The method was also particularly effective at reducing noise variants to yield a smaller set of variants.

MSIpred is an SVM algorithm constructed to detect microsatellite instability (MSI), a condition associated with a high degree of polymorphisms associated with several types of tumors (Wang and Liang 2018). MSIpred uses mutational load data created from whole-exome data. While MSIpred's particular goal is classification, a RF process can be used to determine which features are important.

Not so different after all

One of the important things to note about the methods that were described above is that though they may seem quite different, multiple connections can be drawn between these seemingly disparate algorithms. They are indeed working to solve similar problems. For example, both the SVM kernel functions and deep learning try to tackle the problem of high-dimensional data by reducing the dimensionality to a more palatable linear problem. Obviously, this is done in different ways—SVM kernels embed the data in a space of infinite dimensions where linear separation can be performed and deep learning "learned" to approximate a linear boundary—but the same problem is being solved in both cases. Further RF can be thought of as an adaptive *k*-NN (Lin and Jeon 2006) by considering the distance between data points as the proportion of shared terminal nodes. In this sense, the terminal node essentially becomes the "neighborhood" seen in *k*-NNs. This shows how not only these methods are approaching the similar problems, but also the interconnectedness of many of these algorithms.

Machine selection and parameter tuning

The vast number of ML methods can seem overwhelming, especially to geneticists that are not experts in ML. It is difficult to know which method might be appropriate for a given data set. Furthermore, many ML algorithms require further input than just genetic data, including parameters that need to be tuned and optimized, often by a trial-and-error approach, such as the k term in k-NN. Again, this can be a daunting task to ML non-experts.

One strategy is to run multiple different machines on the same data and then incorporate that data into a new machine. This is the theory behind methods such as synthetic random forest (Ishwaran and Malley 2014) quantitative (regression), the similarity-binning averaging approach (Bella et al. 2013), and optimal crowd (Battogtokh et al. 2017). Synthetic random forests (SRFs) operate under the assumption that differing data might be better fit by a different number of terminal nodes; though tuning this by hand is not feasible. SRFs work by running multiple RFs of varying terminal node size and calculating the predicted value of each RF (the synthetic feature). The synthetic features are then placed into a new RF with the original features (the SRF); SRFs outperform the traditional RFs and optimized RFs (Ishwaran and Malley 2014). The similarity-binning average approach looks to calibrate a model by first running a given model and obtaining the estimated probabilities associated with each dataset, with the estimated probabilities combined with each instance creating a new dataset. The model is then run a second time, and the probabilities from the second run of the model are then placed with the most similar instances (usually determined by *k*-NN) from the first run, creating a bin; with the probability of classification just being the average of the bins. This method has been shown to be empirically better than other calibration methods (Bella et al. 2013). Optimal crowd takes predictions from a family of machines (like RFs and SVMs) that analyzed the same binary data. Using these multiple predictions on the same data allows the machines in the optimal crowd to learn from each other and make a new classification. Optimal crowd has been shown to be at least as good as the best machine in the family (Battogtokh et al. 2017). None of these methods offer feature importance metrics currently, however. We note that there is no one machine that is best for all datasets and problems. Thus, the choice of machines is very difficult, since there is no best machine for all circumstances. This dilemma is indeed the motivating factor for many autoML approaches described below.

Another approach is automated machine learning (autoML). This is a relatively new field that seeks to automate the parameter selection processes, taking the burden

off the user. Thus, instead of requiring the analyst to tune parameters or models, autoML essentially does this heavy lifting for you by building an ML pipeline that optimizes models by model selection, parameter tuning, etc. (Le et al. 2020). Most autoML models focus primarily on prediction, but Tree-based Pipeline Optimization (TPOT), a GP-based autoML method does generate permutation-derived feature importance scores (Orlenko et al. 2020). TPOT has been tested on genetic data, including the evaluation biomarkers for the prediction of heart disease (Chirinos et al. 2020). Currently, autoML methods take a lot of computational power to run, and thus have been confined to relatively small sets of features (Le et al. 2018, 2020). Recent developments in TPOT have increased scalability in large data sets, including a feature selection set that allows for the specification of features into subsets (Le et al. 2020). TPOT also now allows for covariate adjustment, which drastically improved feature importance scores by eliminating false positives in a gene expression study (Manduchi et al. 2020).

Conclusion

Machine learning approaches have greatly increased our ability in genetics to analyze complex data sets, as well as ask more intricate questions. ML approaches in genetics have been mostly commonly used to elicit good predictions, for instance to identify cancer patients through RNA-seq (Shen et al. 2020). However, in genetics feature importance is also a critical field. Often, it is just as important to know what makes a good prediction as it is to get a good prediction by blindly feeding data into a black box. Feature importance allows for ranking of particular features, whether it be for identifying epistatic interaction (Holzinger et al. 2015; Szymczak et al. 2016) or simply determining which features used by a machine are responsible for the best prediction (Ioannidis et al. 2016; Sundaram et al. 2018). While we have primarily discussed feature importance as a way of identifying features that most affect a prediction, we note that importance metrics could be used in the reverse way, meaning that features that are deemed important for poorly performing classifiers might themselves be less informative.

As noted in this review, there are numerous genetic programs that give feature importance scores that can answer a variety of questions, including variant pathogenicity and epistatic interactions. Many of these programs are built using variants of popular, well-known machines like random forest, artificial neural networks, deep learning, and support vector machines. Newer methods, like the autoML approaches (Le et al. 2020), decrease the expertise needed to run some of these complex machines, by tuning parameters automatically instead of requiring user input (though interpretation of results may still require an understanding of the machine). It is clear that genetic data are only to increase in complexity and quantity, so the importance of novel ML approaches will only increase in the coming years. Feature importance metrics for these methods will be critical, as it allows for the researcher to not only identify important variables for prediction, but to see what is happening within the black box of many of these algorithms.

Acknowledgements This work was funded in part by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health.

Author contributions AMM wrote the manuscript. ERH, JDM, and JEBW revised the manuscript and provided additional guidance.

Funding This work was funded in part by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health.

Availability of data and materials This is a review article, so there are no original data associated with it.

Code availability This is a review article, so there is no unique code associated with this work.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethics approval There was no ethics approval needed for this review article.

Consent to participate This work contains no human subjects.

Consent for publication This work contains no human subjects.

References

Abo Alchamlat S, Farnir F (2017) KNN-MDR: a learning approach for improving interactions mapping performances in genome wide association studies. BMC Bioinform 18:184. https://doi.org/10.1186/s12859-017-1599-7

Abu Alfeilat HA, Hassanat ABA, Lasassmeh O, Tarawneh AS, Alhasanat MB, Eyal Salman HS, Prasath VBS (2019) Effects of

distance measure choice on *K*-nearest neighbor classifier performance: a review. Big Data 7:221–248. https://doi.org/10.1089/big.2018.0175

Altmann A, Toloşi L, Sander O, Lengauer T (2010) Permutation importance: a corrected feature importance measure. Bioinformatics 26:1340–1347. https://doi.org/10.1093/bioinformatics/btq134

Arloth J, Eraslan G, Andlauer TFM, Martins J, Iurato S, Kühnel B, Waldenberger M, Frank J, Gold R, Hemmer B, Luessi F, Nischwitz S, Paul F, Wiendl H, Gieger C, Heilmann-Heimbach S, Kacprowski T, Laudes M, Meitinger T, Peters A, Rawal R, Strauch K, Lucae S, Müller-Myhsok B, Rietschel M, Theis FJ, Binder EB, Mueller NS (2020) DeepWAS: Multivariate genotype-phenotype associations by directly integrating regulatory information using deep learning. PLoS Comput Biol 16:e1007616. https://doi.org/10.1371/journal.pcbi.1007616

Basile AO, Ritchie MD (2018) Informatics and machine learning to define the phenotype. Expert Rev Mol Diagn 18:219–226. https://doi.org/10.1080/14737159.2018.1439380

Battogtokh B, Mojirsheibani M, Malley J (2017) The optimal crowd learning machine. BioData Min 10:16. https://doi.org/10.1186/s13040-017-0135-7

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana MJ (2013) On the effect of calibration in classifier combination. Appl Intell 38:566–585

Boulesteix AL, Bender A, Lorenzo Bermejo J, Strobl C (2012) Random forest Gini importance favours SNPs with large minor allele frequency: impact, sources and recommendations. Brief Bioinform 13:292–304. https://doi.org/10.1093/bib/bbr053

Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324

Cai YD, Liu XJ, Chou KC (2003) Prediction of protein secondary structure content by artificial neural network. J Comput Chem 24:727–731. https://doi.org/10.1002/jcc.10222

Carter H, Douville C, Stenson PD, Cooper DN, Karchin R (2013) Identifying Mendelian disease genes with the variant effect scoring tool. BMC Genom 14(Suppl 3):S3. https://doi.org/10.1186/1471-2164-14-s3-s3

Chirinos JA, Orlenko A, Zhao L, Basso MD, Cvijic ME, Li Z, Spires TE, Yarde M, Wang Z, Seiffert DA, Prenner S, Zamani P, Bhattacharya P, Kumar A, Margulies KB, Car BD, Gordon DA, Moore JH, Cappola TP (2020) Multiple plasma biomarkers for risk stratification in patients with heart failure and preserved ejection fraction. J Am Coll Cardiol 75:1281–1295. https://doi.org/10.1016/j.jacc.2019.12.069

Curtis D (2007) Comparison of artificial neural network analysis with other multimarker methods for detecting genetic association. BMC Genet 8:49. https://doi.org/10.1186/1471-2156-8-49

de Oliveira FC, Borges CC, Almeida FN, de Silva FF, da Silva-Verneque R, da Silva MV, Arbex W (2014) SNPs selection using support vector regression and genetic algorithms in GWAS. BMC Genom 15(Suppl 7):S4. https://doi.org/10.1186/1471-2164-15-s7-s4

Degenhardt F, Seifert S, Szymczak S (2019) Evaluation of variable selection methods for random forests and omics data sets. Brief Bioinform 20:492–503. https://doi.org/10.1093/bib/bbx124

Devroye L, Gyorfi L, Krzyzak A, Lugosi G (1994) On the strong universal consistency of nearest neighbor regression function estimates. Ann Stat 22:1371–1385

Efron B (2020) Prediction, estimation, and attribution. J Am Stat Assoc 115:636–655. https://doi.org/10.1080/01621459.2020.1762613

Eraslan G, Avsec Ž, Gagneur J, Theis FJ (2019) Deep learning: new computational modelling techniques for genomics. Nat Rev Genet 20:389–403. https://doi.org/10.1038/s41576-019-0122-6

Ghoussaini M, Mountjoy E, Carmona M, Peat G, Schmidt EM, Hercules A, Fumis L, Miranda A, Carvalho-Silva D, Buniello A, Burdett T, Hayhurst J, Baker J, Ferrer J, Gonzalez-Uriarte A, Jupp S, Karim MA, Koscielny G, Machlitt-Northen S, Malangone C, Pendlington ZM, Roncaglia P, Suveges D, Wright D, Vrousgou O, Papa E, Parkinson H, MacArthur JAL, Todd JA, Barrett JC, Schwartzentruber J, Hulcoop DG, Ochoa D, McDonagh EM, Dunham I (2021) Open targets genetics: systematic identification of trait-associated genes using large-scale genetics and functional genomics. Nucleic Acids Res 49:D1311-d1320. https://doi.org/10.1093/nar/gkaa840

Greene CS, Penrod NM, Kiralis J, Moore JH (2009) Spatially uniform relieff (SURF) for computationally-efficient filtering of gene-gene interactions. BioData Min 2:5. https://doi.org/10.1186/1756-0381-2-5

Greenside P, Shimko T, Fordyce P, Kundaje A (2018) Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. Bioinformatics 34:i629–i637. https://doi.org/10.1093/bioinformatics/bty575

Guo L, Wang Z, Du Y, Mao J, Zhang J, Yu Z, Guo J, Zhao J, Zhou H, Wang H, Gu Y, Li Y (2020) Random-forest algorithm based biomarkers in predicting prognosis in the patients with hepatocellular carcinoma. Cancer Cell Int 20:251. https://doi.org/10.1186/s12935-020-01274-z

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422. https://doi.org/10.1023/A:1012487302797

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York

Holzinger ER, Dudek SM, Frase AT, Pendergrass SA, Ritchie MD (2014) ATHENA: the analysis tool for heritable and environmental network associations. Bioinformatics 30:698–705. https://doi.org/10.1093/bioinformatics/btt572

Holzinger ER, Szymczak S, Dasgupta A, Malley J, Li Q, Bailey-Wilson JE (2015) Variable selection method for the identification of epistatic models. In: Pacific Symposium on Biocomputing, pp 195–206

Holzinger ER, Szymczak S, Malley J, Pugh EW, Ling H, Griffith S, Zhang P, Li Q, Cropp CD, Bailey-Wilson JE (2016) Comparison of parametric and machine methods for variable selection in simulated Genetic Analysis Workshop 19 data. BMC Proc 10:147–152. https://doi.org/10.1186/s12919-016-0021-1

Hu Y, Hase T, Li HP, Prabhakar S, Kitano H, Ng SK, Ghosh S, Wee LJ (2016) A machine learning approach for the identification of key markers involved in brain development from single-cell transcriptomic data. BMC Genom 17:1025. https://doi.org/10.1186/s12864-016-3317-7

Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y, Xu W (2018) Applications of support vector machine (SVM) learning in cancer genomics. Cancer Genom Proteom 15:41–51. https://doi.org/10.21873/cgp.20063

Ioannidis NM, Rothstein JH, Pejaver V, Middha S, McDonnell SK, Baheti S, Musolf A, Li Q, Holzinger E, Karyadi D, Cannon-Albright LA, Teerlink CC, Stanford JL, Isaacs WB, Xu J, Cooney KA, Lange EM, Schleutker J, Carpten JD, Powell IJ, Cussenot O, Cancel-Tassin G, Giles GG, MacInnis RJ, Maier C, Hsieh CL, Wiklund F, Catalona WJ, Foulkes WD, Mandal D, Eeles RA, Kote-Jarai Z, Bustamante CD, Schaid DJ, Hastie T, Ostrander EA, Bailey-Wilson JE, Radivojac P, Thibodeau SN, Whittemore AS, Sieh W (2016) REVEL: an ensemble method for predicting the pathogenicity of rare missense variants. Am J Hum Genet 99:877–885. https://doi.org/10.1016/j.ajhg.2016.08.016

Ishwaran H, Malley JD (2014) Synthetic learning machines. Biodata Min 7:28. https://doi.org/10.1186/s13040-014-0028-y

Janitza S, Celik E, Boulesteix A-L (2018) A computationally fast variable importance test for random forests for high-dimensional data. Adv Data Anal Classif 12:885–915

Jiang R, Tang W, Wu X, Fu W (2009) A random forest approach to the detection of epistatic interactions in case-control studies. BMC Bioinform 10(Suppl 1):S65. https://doi.org/10.1186/1471-2105-10-S1-S65

Kelley DR, Snoek J, Rinn JL (2016) Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res 26:990–999. https://doi.org/10.1101/gr.200535.115

Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, Snoek J (2018) Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res 28:739–750. https://doi.org/10.1101/gr.227819.117

Kircher M, Witten DM, Jain P, O'Roak BJ, Cooper GM, Shendure J (2014) A general framework for estimating the relative pathogenicity of human genetic variants. Nat Genet 46:310–315. https://doi.org/10.1038/ng.2892

Kursa MB, Rudnicki WR (2010) Feature selection with the Boruta package. J Stat Softw 36:1–13

Le TT, Blackwood NO, Taroni JN, Fu W, Breitenstein MK (2018) Integrated machine learning pipeline for aberrant biomarker enrichment (i-mAB): characterizing clusters of differentiation within a compendium of systemic lupus erythematosus patients. AMIA Annu Symp Proc 2018:1358–1367

Le TT, Urbanowicz RJ, Moore JH, McKinney BA (2019) STatistical Inference Relief (STIR) feature selection. Bioinformatics 35:1358–1365. https://doi.org/10.1093/bioinformatics/bty788

Le TT, Fu W, Moore JH (2020) Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics 36:250–256. https://doi.org/10.1093/bioinformatics/btz470

Li J, Malley JD, Andrew AS, Karagas MR, Moore JH (2016) Detecting gene-gene interactions using a permutation-based random forest method. BioData Min 9:14. https://doi.org/10.1186/s13040-016-0093-5

Li J, Jew B, Zhan L, Hwang S, Coppola G, Freimer NB, Sul JH (2019) ForestQC: Quality control on genetic variants from next-generation sequencing data using random forest. PLoS Comput Biol 15:e1007556. https://doi.org/10.1371/journal.pcbi.1007556

Lin Y, Jeon Y (2006) Random forests and adaptive nearest neighbors. J Am Stat Assoc 101:578–590

Liu L, Chen X, Wong KC (2021) Early cancer detection from genome-wide cell-free DNA fragmentation via shuffled frog leaping algorithm and support vector machine. Bioinformatics. https://doi.org/10.1093/bioinformatics/btab236

Lunetta KL, Hayward LB, Segal J, Van Eerdewegh P (2004) Screening large-scale association study data: exploiting interactions using random forests. BMC Genet 5:32. https://doi.org/10.1186/1471-2156-5-32

Malley JD, Malley KG, Pajevic S (2011) Statistical learning for biomedical data. Cambridge University Press

Manduchi E, Fu W, Romano JD, Ruberto S, Moore JH (2020) Embedding covariate adjustments in tree-based automated machine learning for biomedical big data analyses. BMC Bioinform 21:430. https://doi.org/10.1186/s12859-020-03755-4

Mao YF, Yuan XG, Cun YP (2021) A novel machine learning approach (svmSomatic) to distinguish somatic and germline mutations using next-generation sequencing data. Zool Res 42:246–249. https://doi.org/10.24272/j.issn.2095-8137.2021.014

McKinney BA, White BC, Grill DE, Li PW, Kennedy RB, Poland GA, Oberg AL (2013) ReliefSeq: a gene-wise adaptive-K nearest-neighbor feature selection tool for finding gene-gene interactions and main effects in mRNA-Seq gene expression data. PLoS ONE 8:e81527. https://doi.org/10.1371/journal.pone.0081527

Moore JH, Williams SM (2009) Epistasis and its implications for personal genetics. Am J Hum Genet 85:309–320. https://doi.org/10.1016/j.ajhg.2009.08.006

Moore JH, Gilbert JC, Tsai CT, Chiang FT, Holden T, Barney N, White BC (2006) A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. J Theor Biol 241:252–261. https://doi.org/10.1016/j.jtbi.2005.11.036

Moore JH, Asselbergs FW, Williams SM (2010) Bioinformatics challenges for genome-wide association studies. Bioinformatics 26:445–455. https://doi.org/10.1093/bioinformatics/btp713

Nait Saada J, Kalantzis G, Shyr D, Cooper F, Robinson M, Gusev A, Palamara PF (2020) Identity-by-descent detection across 487,409 British samples reveals fine scale population structure and ultra-rare variant associations. Nat Commun 11:6130. https://doi.org/10.1038/s41467-020-19588-x

Nembrini S, König IR, Wright MN (2018) The revival of the Gini importance? Bioinformatics 34:3711–3718. https://doi.org/10.1093/bioinformatics/bty373

Neuditschko M, Khatkar MS, Raadsma HW (2012) NetView: a high-definition network-visualization approach to detect fine-scale population structures from genome-wide patterns of variation. PLoS ONE 7:e48375. https://doi.org/10.1371/journal.pone.0048375

Ng PC, Henikoff S (2003) SIFT: Predicting amino acid changes that affect protein function. Nucleic Acids Res 31:3812–3814

Nicodemus KK (2011) Letter to the editor: on the stability and ranking of predictors from random forest variable importance measures. Brief Bioinform 12:369–373. https://doi.org/10.1093/bib/bbr016

Olden JD, Jackson DA (2002) Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. Ecol Model 154:135–150

Orlenko A, Moore JH (2021) A comparison of methods for interpreting random forest models of genetic association in the presence of non-additive interactions. BioData Min 14:9. https://doi.org/10.1186/s13040-021-00243-0

Orlenko A, Kofink D, Lyytikäinen LP, Nikus K, Mishra P, Kuukasjärvi P, Karhunen PJ, Kähönen M, Laurikka JO, Lehtimäki T, Asselbergs FW, Moore JH (2020) Model selection for metabolomics: predicting diagnosis of coronary artery disease using automated machine learning. Bioinformatics 36:1772–1778. https://doi.org/10.1093/bioinformatics/btz796

Poplin R, Chang PC, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, Gross SS, Dorfman L, McLean CY, DePristo MA (2018) A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol 36:983. https://doi.org/10.1038/nbt.4235

Rentzsch P, Witten D, Cooper GM, Shendure J, Kircher M (2019) CADD: predicting the deleteriousness of variants throughout the human genome. Nucleic Acids Res 47:D886–D894. https://doi.org/10.1093/nar/gky1016

Schölkopf B, Tsuda K, Vert J-P (2003) Kernel methods in computational biology. MIT Press

Schwarz DF, Konig IR, Ziegler A (2010) On safari to random jungle: a fast implementation of random forests for high-dimensional data. Bioinformatics 26:1752–1758. https://doi.org/10.1093/bioinformatics/btq257

Shen Y, Lai Y, Xu D, Xu L, Song L, Zhou J, Song C, Wang J (2020) Diagnosis of thyroid neoplasm using support vector machine algorithms based on platelet RNA-seq. Endocrine. https://doi.org/10.1007/s12020-020-02523-x

Shi T, Horvath S (2006) Unsupervised learning with random forest predictors. J Comput Graph Stat 15:118–138

Shihab HA, Gough J, Cooper DN, Stenson PD, Barker GL, Edwards KJ, Day IN, Gaunt TR (2013) Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Hum Mutat 34:57–65. https://doi.org/10.1002/humu.22225

Strobl C, Boulesteix A-L, Kneib T, Augustin T, Zeileis A (2008) Conditional variable importance for random forests. BMC Bioinform 9:1–11

Sundaram L, Gao H, Padigepati SR, McRae JF, Li Y, Kosmicki JA, Fritzilas N, Hakenberg J, Dutta A, Shon J, Xu J, Batzoglou S, Li X, Farh KK (2018) Predicting the clinical impact of human mutation with deep neural networks. Nat Genet 50:1161–1170. https://doi.org/10.1038/s41588-018-0167-z

Szymczak S, Biernacka JM, Cordell HJ, González-Recio O, König IR, Zhang H, Sun YV (2009) Machine learning in genome-wide association studies. Genet Epidemiol 33(Suppl 1):S51–S57. https://doi.org/10.1002/gepi.20473

Szymczak S, Holzinger E, Dasgupta A, Malley JD, Molloy AM, Mills JL, Brody LC, Stambolian D, Bailey-Wilson JE (2016) r2VIM: a new variable selection method for random forests in genome-wide association studies. BioData Min 9:7. https://doi.org/10.1186/s13040-016-0087-3

Teixeira PL, Wei WQ, Cronin RM, Mo H, VanHouten JP, Carroll RJ, LaRose E, Bastarache LA, Rosenbloom ST, Edwards TL, Roden DM, Lasko TA, Dart RA, Nikolai AM, Peissig PL, Denny JC (2017) Evaluating electronic health record data sources and algorithmic approaches to identify hypertensive individuals. J Am Med Inform Assoc 24:162–171. https://doi.org/10.1093/jamia/ocw071

Tong DL, Schierz AC (2011) Hybrid genetic algorithm-neural network: feature extraction for unpreprocessed microarray data. Artif Intell Med 53:47–56. https://doi.org/10.1016/j.artmed.2011.06.008

Tong DL, Boocock DJ, Dhondalay GK, Lemetre C, Ball GR (2014) Artificial neural network inference (ANNI): a study on gene-gene interaction for biomarkers in childhood sarcomas. PLoS ONE 9:e102483. https://doi.org/10.1371/journal.pone.0102483

Turner SD, Dudek SM, Ritchie MD (2010) ATHENA: a knowledge-based hybrid backpropagation-grammatical evolution neural network algorithm for discovering epistasis among quantitative trait Loci. BioData Min 3:5. https://doi.org/10.1186/1756-0381-3-5

Tyanova S, Albrechtsen R, Kronqvist P, Cox J, Mann M, Geiger T (2016) Proteomic maps of breast cancer subtypes. Nat Commun 7:10259. https://doi.org/10.1038/ncomms10259

Wang C, Liang C (2018) MSIpred: a python package for tumor microsatellite instability classification from tumor mutation annotation data using a support vector machine. Sci Rep 8:17546. https://doi.org/10.1038/s41598-018-35682-z

Wang Q, Yu C (2020) Expression profiling of small intestinal neuroendocrine tumors identified pathways and gene networks linked to tumorigenesis and metastasis. Biosci Rep. https://doi.org/10.1042/bsr20193860

Wang M, Tai C, Weinan E, Wei L (2018) DeFine: deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional noncoding variants. Nucleic Acids Res 46:e69. https://doi.org/10.1093/nar/gky215

Wilhelm T (2014) Phenotype prediction based on genome-wide DNA methylation data. BMC Bioinform 15:193. https://doi.org/10.1186/1471-2105-15-193

Winham SJ, Freimuth RR, Biernacka JM (2013) A weighted random forests approach to improve predictive performance. Stat Anal Data Min 6:496–505. https://doi.org/10.1002/sam.11196

Wright MN, Ziegler A (2017) ranger: a fast implementation of random forests for high dimensional data in C plus plus and R. J Stat Softw 77:1–17. https://doi.org/10.18637/jss.v077.i01

Xu G, Zhang M, Zhu H, Xu J (2017) A 15-gene signature for prediction of colon cancer recurrence and prognosis based on SVM. Gene 604:33–40. https://doi.org/10.1016/j.gene.2016.12.016

Yang W, Charles GuC (2014) Random forest fishing: a novel approach to identifying organic group of risk factors in genome-wide association studies. Eur J Hum Genet 22:254–259. https://doi.org/10.1038/ejhg.2013.109

Zhou J, Troyanskaya OG (2015) Predicting effects of noncoding variants with deep learning-based sequence model. Nat Methods 12:931–934. https://doi.org/10.1038/nmeth.3547

Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG (2018) Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat Genet 50:1171–1179. https://doi.org/10.1038/s41588-018-0160-6