OXFORD

Briefings in Functional Genomics, 19(1), 2020, 40–48

doi: 10.1093/bfgp/elz036
Advance Access Publication Date: 23 December 2019
Review paper

Machine learning and its applications in plant molecular studies

Shanwen Sun, Chunyu Wang, Hui Ding and Quan Zou

Corresponding author: Quan Zou, Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mail: zouquan@nclab.net

Abstract

The advent of high-throughput genomic technologies has resulted in the accumulation of massive amounts of genomic information. However, biologists are challenged with how to effectively analyze these data. Machine learning can provide tools for better and more efficient data analysis. Unfortunately, because many plant biologists are unfamiliar with machine learning, its application in plant molecular studies has been restricted to a few species and a limited set of algorithms. Thus, in this study, we provide the basic steps for developing machine learning frameworks and present a comprehensive overview of machine learning algorithms and various evaluation metrics. Furthermore, we introduce sources of important curated plant genomic data and R packages to enable plant biologists to easily and quickly apply appropriate machine learning algorithms in their research. Finally, we discuss current applications of machine learning algorithms for identifying various genes related to resistance to biotic and abiotic stress. Broad application of machine learning and the accumulation of plant sequencing data will advance plant molecular studies.

Key words: supervised machine learning; unsupervised machine learning; evaluation metrics; plants; genomics

Introduction

The advent of high-throughput sequencing technologies has produced several large-scale data sets. This enormous amount of information enables biologists to explore topics that were once difficult or impossible to investigate, such as associations between microRNA and certain diseases, the causes of vascular inflammation and atherosclerosis in humans [1–3] and stress breeding in plants [4]. However, many challenges have also emerged. For example, the European Bioinformatics Institute now stores 273 petabytes of raw molecular data on humans, plants and animals (https://www.ebi.ac.uk/). These data are characterized by high dimensionality, high uncertainty (such as missing values, measurement error and data entry errors) and nonindependence and are impossible to process and analyze using traditional regression frameworks. Analyzing such large data sets requires modern learning approaches. Machine learning provides techniques for addressing high dimensionality and missing values while simultaneously exploiting the information hidden in large databases.

Machine learning is a field of computer science that teaches machines to automatically extract important information from accumulated examples to improve predictions and find associations and patterns in data [5–11]. Machine learning is generally grouped into two major kinds: supervised and unsupervised. Supervised learning uses labeled data (i.e. the output values of examples are known). Unsupervised learning, in contrast, uses unlabeled data. Machine learning has great potential in biology at the population, individual and genetic levels and has

Shanwen Sun received his PhD from the University of Bayreuth in Germany. He is now a postdoctoral fellow at the Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China.
Chunyu Wang received his PhD from the Harbin Institute of Technology in China. He is an associate professor in the School of Computer Science and Technology, Harbin Institute of Technology.
Hui Ding received her PhD from Inner Mongolia University in China. She is an associate professor in the Center for Informational Biology, University of Electronic Science and Technology of China.
Quan Zou received his PhD from the Harbin Institute of Technology in China. He is a professor in the Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China.

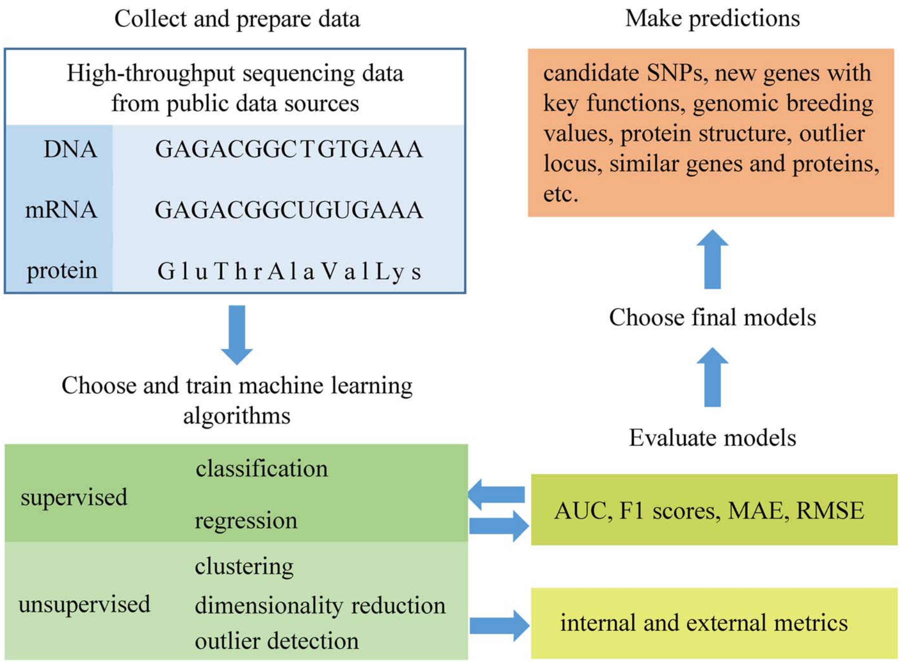
Collect and prepare data
High-throughput sequencing data from public data sources
DNA GAGACGGCTGTGAAA
mRNA GAGACGGCUGUGAAA
protein GluThrAlaValLys
Choose and train machine learning algorithms
supervised
classification
regression
unsupervised
clustering
dimensionality reduction
outlier detection
Evaluate models
AUC, F1 scores, MAE, RMSE
internal and external metrics
Choose final models
Make predictions
candidate SNPs, new genes with key functions, genomic breeding values, protein structure, outlier locus, similar genes and proteins, etc.

Figure 1. The basic steps for building a machine learning framework. The first step is to collect data from public sources of curated sequencing data based on the research questions. The data need to be preprocessed to remove duplicates, eradicate errors and address missing values. The second step is to select and train machine learning algorithms depending on the properties of the data and the questions at hand. Supervised algorithms can be used to make predictions with labeled data (classification for category data, regression for continuous data). Unsupervised algorithms can be used to cluster data, reduce dimensionality and detect outliers. See Table 1 for representative algorithms and R packages for implementing these algorithms. After they are trained, models can be evaluated based on various metrics to tune model hyperparameters and later to choose the final models. Widely used metrics for supervised learning are the AUC and the F1 score for classification, and the RMSE and MAE for regression. External and internal metrics can be used to assess the performance of unsupervised learning. The final step in a typical machine learning framework in plant molecular studies is to make predictions about candidate genes responsive to disease or other stress, breeding values, and so on.

demonstrated its power in the analysis of massive and complex sets of data on animals. For example, machine learning approaches have been used to understand the transcription of genomic information of *Drosophila* into cellular and developmental programs [12, 13] and to explore functions of the human genome and the regulation of human genes [14–16]. In plants, machine learning has been used to identify stress-responsive genes in *Arabidopsis* [17] and to breed stress-resistant crops [4].

Despite these applications, machine learning remains underexploited and sometimes misunderstood and misused by plant scientists because of both the complexity of life systems and researchers' own unfamiliarity with machine learning algorithms. Thus, in this review, we present the basic steps for building machine learning systems and describe the main machine learning algorithms and metrics for tuning model hyperparameters and evaluating model performance. In addition, we introduce important publicly available, curated plant genomic databases and describe packages in the R environment [18] that can help researchers quickly and easily implement machine learning algorithms. Finally, we discuss specific applications of machine learning algorithms to identify genes related to stress and disease resistance in plants.

The basic steps for building machine learning tools

Collect and prepare data

Collecting data is the first step in developing a machine learning framework (Figure 1). A large amount of genomic data is available to the public and updated daily, which facilitates the advancement of biological research. However, these data come from various sources and may be incorrect, incomplete or duplicated. It is extremely important to use reliable and professionally curated data sources when implementing machine learning. Below we introduce several important curated public databases that can help plant scientists to initialize or carry out their studies.

The Reference Sequence (RefSeq) database, created by National Center for Biotechnology Information (NCBI), provides well-annotated and nonredundant sequence data [19]. It contains more than 60 million reference sequence data points from different kinds of organisms marked by seven status, i.e. wgs, model, predicted, provisional, inferred, validated and reviewed [19]. The final two classifications indicate that sequences have been curated by NCBI staff or collaborating groups [19]. A total of 71 762 out of 8 144 248 plant sequences are professionally annotated in the RefSeq database [19], which suggests a lag between the collection of sequence data and their annotation and organization by RefSeq.

Other efforts are targeted at curating data from GenBank to comply with specific requirements and community needs. For example, Gramene hosts curated and assembled sequence data on crops and other model plants [20]. The current release includes reference genomes from 61 species [20]. Phytozome, developed and maintained by the Department of Energy's Joint Genome Institute, is a hub for analyzing Viridiplantae genomics [21]. It now provides 93 integrated and annotated genomes of 82 plant species together with a series of tools for querying, viewing and analyzing sequence data, such as BLAST, Jbrowse, PhytoMine and BioMart [21]. The Plant Resistance Genes database (PRGdb) is a curated database of pathogen resistance genes (PRGs) of 268 species [22]. A total of 35 species have reference PRGs and 99 species have annotated PRGs [22].

Some taxon- and species-specific databases also exist to provide researchers with even more in-depth, accurate and up-to-date genomic information. The Sol Genomics Network (SGN) hosts genomic data on many important crops of the Solanaceae family [23]. The Legume Information System (LIS) provides integrated genomic information on 22 legume species, including two important model plants, *Medicago truncatula* and *Lotus japonicus* [24]. The Arabidopsis Information Resource (TAIR) has information about a widely used model plant, *Arabidopsis thaliana*, including complete genomic sequences, genome maps, gene structure and gene expression data [25]. The Maize Genetics and Genomics Database (MaizeGDB) stores high-quality sequence data for many inbred strains of maize [26]. The Information Commons for Rice (IC4R) provides an updated, curated and integrated database of rice genomes based on contributions from the community [27]. The International Wheat Genome Sequencing Consortium (IWGSC) now hosts reference sequences of wheat [28].

Choose an algorithm or algorithms

Many algorithms have been created in the development of machine learning. Different algorithms fit different tasks and data types. Choosing the right algorithm or algorithms is fundamental to using machine learning tools (Figure 1). As described above, supervised machine learning uses labeled output in the training data set. Its task is to make predictions about unlabeled output based on new inputs. It can be further categorized as classification or regression depending on whether discrete or continuous outputs are mapped. Representative classification algorithms include penalized logistic regression, support vector machine (SVM) [29–32], random forest [33–36] and neural network (Table 1). Common regression algorithms are linear regression, random forest regression, support vector regression, ridge regression, lasso regression and elastic net regression (Table 1).

Penalized logistic regression is an extension of traditional logistic regression that is usually preferred for fitting binary response data. It uses all available variables and imposes a penalty through regularization on the complexity of the logistic model and shrinks the coefficients of trivial variables toward zero. Its use is advocated in genetic association analysis to identify important single-nucleotide polymorphisms (SNPs) [37] and model complex gene–gene and gene–environment interactions to identify influential genes and interactions for common diseases [38].

SVM can be used for both classification and regression. It is used to construct a hyperplane that has the greatest distance to the nearest examples in the training data set and thus creates the biggest margin between different classes and provides good separation. It has broad applications in bioinformatics, such as predicting subcellular localization of proteins [39–41], identifying translation initiation sites that start to code for proteins [42], predicting the functions of proteins [43, 44], identifying RNA modification sites [45, 46], analyzing gene expression profiling data [47] and recognizing promoters [48] and alternative spliced exons [49].

Random forest is an ensemble method that is used to construct an entire forest of random decision trees trained with the 'bagging' method and merge them to gain better predictive results. It overcomes the primary drawback of decision trees (i.e. overfitting on the training data set). Similar to SVM, random forest can be used for both classification and regression. It has been utilized in different areas of genomics as well, including recognizing associations between SNPs and traits [50], identifying protein DNA-binding sites [51], detecting epistatic interactions between genetic variants [52] and predicting Alzheimer's disease [53].

Neural network imitates the neuronal structure of the human brain with interconnected nodes organized into layers. Training data are first fed into the input layer, then processed through one or more succeeding hidden layers and finally linked to the output layer. The model is usually trained using backpropagation. Neural network has outperformed state-of-the-art techniques in many fields, such as image classification, computational chemistry and high-energy physics [54]. In genomics, it has been used to analyze gene expression data [55], to annotate enhancers and promoters [56] and to identify translation initiation sites [57].

Linear regression is prevalent in genomics for its simplicity and interpretability. For example, Larsson *et al.* [58] used linear regression to assess the effects of the core promoter element on transcriptional bursting. Gorlov *et al.* [59] used linear regression to predict gene mutations in cancer samples based on gene characteristics. However, when used to analyze highly dimensional data, linear regression often has high prediction error and high variance [60]. Regularization algorithms, such as ridge regression, lasso regression and elastic net regression, are usually used to solve the problem of high dimensionality. These methods show overall better performance than linear regression [61] and have gained in popularity in genome-wide association studies. Researchers have used them to predict genomic breeding values for animals [62] and plants [63] based on all molecular markers and to identify SNPs corresponding to traits of interest [64].

Typical tasks of unsupervised learning are clustering, dimensionality reduction and outlier detection (Table 1). The aim of clustering is to group examples into partitions that contain similar characteristics based on some distance or similarity metrics. Different clustering algorithms may fit different needs and data types. Centroid-based algorithms, such as K-means, one of the most commonly used clustering algorithms, produce K centroids as the center of corresponding clusters that have a minimum distance to other examples in the same clusters. They are simple and efficient but relatively sensitive to outliers and sometimes may reach local optimum [65]. Density-based algorithms, such as density-based spatial clustering of applications with noise (DBSCAN) and mean shift, define a cluster as an area with a high density of examples. Examples in the sparse area are assigned as outliers. An advantage of density-based clustering is that the cluster shape can be arbitrary [65]. Distribution-based algorithms, such as expectation–maximization (EM), assume that examples in the data are sampled from different Gaussian distributions and assign each example to a cluster based on probability. It is supported by traditional statistics but also assumes a Gaussian distribution that may not be correct for many real data [65]. Hierarchical clustering is used to create a hierarchical tree that describes the relationships among examples. More closely related examples appear closer on the tree. This method is well suited to analyzing hierarchical data or data with an arbitrary shape [65]. Overall, clustering has broad applications in genomics, such as identifying similar genes [66], exploring gene regulatory networks [67] and aiding in genotype calling [68].

Dimensionality reduction is crucial for analyzing highly dimensional high-throughput sequencing data. It enables researchers to select the most relevant features or genes to characterize a system, enhance model performance and reduce

Table 1. Categories of machine learning, their tasks, representative algorithms and examples of R packages for implementation

	Categories of machine learning	Tasks	Representative algorithms	Examples of R packages
Supervised	Classification	Make a prediction about an unlabeled binary or multiclass output in response to an input	Penalized logistic regression, naive Bayes classifier, nearest neighbor, SVM, decision tree, random forest, neural network	glmnet, naivebayes, class, e1071, tree, caret, randomForest, neuralnet
	Regression	Assess relations between inputs and outputs, and make a prediction about unlabeled continuous output in response to an input	Linear regression, random forest regression, support vector regression, ridge regression, lasso regression, elastic net regression	stats, caret, tree, randomForest, e1071, ridge, glmnet
Unsupervised	Clustering	Find a partition to group the observed data so that similar examples are in the same group and dissimilar to those in other groups, without explicit labels implying an anticipated partition	K-means clustering, mean-shift clustering, DBSCAN, EM clustering using Gaussian mixture models, balanced iterative reducing and clustering using hierarchies	stats, MeanShift, DBSCAN, GMCM, stream, e1071
	Dimensionality reduction	Transform highly dimensional data into a space with fewer, more meaningful dimensions	PCA, kernel PCA	stats, MASS, kernlab
	Outlier detection	Identify unexpected observations that differ from larger collections of such observations	Index-based algorithm, nested-loop algorithm, cell-based algorithm, K-nearest neighbor, local outlier factor, connectivity-based outlier factor	OutlierDetection, DDoutlier, DBSCAN

time complexity [69–73]. Principal components analysis (PCA) is a classic method for reducing dimensionality by transforming an entire highly dimensional data set to a few derived features with minimal loss of variation. Results from PCA are used either for data visualization [74] or as intermediate results for further clustering [75] and regression [76] analyses.

Finally, outlier detection is another important application of unsupervised learning in molecular studies. An outlier is an example that behaves differently from the rest of the data according to some criteria, such as distance or density estimate to the nearest neighbor. Algorithms for outlier detection (Table 1) have been used to identify the *outlier locus* under selection [77] and to identify differentially expressed genes [78].

Train and evaluate the models

The R Project for Statistical Computing provides a powerful environment for manipulating, calculating and visualizing data and ample tools for implementing machine learning algorithms [18]. Many well-developed packages exist to help researchers train and evaluate their models and make predictions (Table 1). Breiman's random forest algorithm can be implemented in randomForest [79]. caret provides a set of tools for training classification and regression algorithms and building predictive models, including functions for preprocessing and splitting data, selecting features and resampling [80]. Regularized linear regression models, such as lasso and elastic net generalized linear models, can be trained using glmnet [81]. e1071 has functions for implementing SVM and many clustering algorithms [82]. The stats package can perform basic regressions, clustering and PCA [18]. Various algorithms for detecting outliers, such as distance-based outlier detection and density-based outlier detection, are included in OutlierDetection [83]. The reference manuals for the packages provide more details about implementing specific functions.

To train and evaluate different supervised learning algorithms, one must first split the data into training, validation and test data sets. Examples in the training data set are used to fit and train different models. After each model is trained, examples in the validation data set are used to assess the predictive accuracy of the trained model and tune its hyperparameters to enhance performance (Figure 1).

Several metrics exist for assessing the performance of models and improving hyperparameters. The choice of metric depends on the algorithms that are trained, although in practice multiple metrics are used simultaneously. For classification, classification accuracy straightforwardly provides the ratio of the number of correct predictions to the total number of predictions. However, it is not fit for use with class-imbalanced data sets [84]. The confusion matrix, the N × N matrix, shows a complete view of the performance of a model. Four items are contained in the confusion matrix: true positives (positives that are correctly predicted as positives), true negatives (negatives that are correctly predicted as negatives), false positives (negatives that are wrongly predicted as positives) and false negatives (positives that are wrongly predicted as negatives). The confusion matrix forms the basis for calculating other types of metrics. Precision is the ratio of the number of true positives to the total number of all predicted positives (i.e. true positives + false positives). Recall is the ratio of the number of true positives to the total number of actual positives (i.e. true positives + false negatives). The F1 score is the harmonic mean of the precision and recall of the model; it allows the researcher to assess the performance of the model based on the balance between precision and recall. The area under the ROC curve (AUC) is another popular metric. The receiver operating characteristic (ROC) curve is a graph created by plotting the true positive rate (a synonym of recall) against the false positive rate (the ratio of the number of false positives to the total number of actual negatives; i.e. false positives + true negatives) at various classification thresholds from zero to one. The AUC calculates the area under the ROC curve indicating the model's capacity to avoid false classification [85]. The larger the AUC, the better the performance of the model.

The root mean square error (RMSE) is one of the most widely used metrics for evaluating regression algorithms. It is the square root of the average of squared differences between observed values and predicted values. RMSE is greatly affected by errors because data are squared before being averaged. Other metrics include the mean absolute error (MAE), R^2, and adjusted R^2. MAE is the average of absolute differences between observed values and predicted values, which is relatively stable in the presence of large error. R^2 gives an intuitive assessment of model performance (i.e. the percentage of the variance in the dependent variable that can be explained by the model). Increasing R^2 by adding new features to the model, however, may lead to overfitting. Adjusted R^2 can penalize models for adding features that do not improve their performance.

Cross-validation is an alternative to splitting the data into three partitions when the sample size is small [86–88]. In cross-validation, data are divided into training and test data sets. Examples in the training data set are repeatedly split into a training data set to train the model and a validation data set to assess the performance of the model for later tuning of model hyperparameters. K-fold cross-validation is the most commonly used method of cross-validation [89–91]. In this technique, training samples are randomly partitioned into K subsamples of roughly equal size. One subsample is iteratively left out as the validation data, and the remaining K – 1 subsamples are used to train the model. In total, the process is repeated K times; every example in the data is trained K – 1 times and validated once. The estimated performance of the model is averaged over all K results from the validation data. As a rule of thumb, K = 5 or K = 10 is usually used. A variation on cross-validation is stratified K-fold cross-validation, which rearranges the data so that each subsample is a good representation of the complete data. This method is preferred for largely imbalanced data.

After finding the model of each algorithm with the best set of hyperparameter values, the researcher uses the test data set to reevaluate the model to obtain reliable and objective estimates of the performance of different algorithms. Final models of sufficient accuracy can be used further to make predictions (Figure 1).

Unsupervised learning uses unlabeled data, and whole data are usually used to train the model; therefore, the evaluation of model performance is often subjective and subject specific. Metrics for validating clustering are external validation and internal validation. External validation uses priori known clustering structures, such as known gene families, to measure the degree of correspondence between the true class labels and the cluster labels categorized by the model [92]. Commonly used external validation metrics are purity, the F1 score, and normalized mutual information (NMI). Purity evaluates the extent to which clusters contain a single class [92]. The F1 score, as discussed previously, provides a balance between precision and recall. NMI assesses how much information is shared between a cluster and the priori known clustering [93]. In general, higher purity and NMI indicate better clustering [92]. Internal validation is useful for evaluating competitive algorithms when clustering in a data set is unknown. It estimates model performance using quantities and features inherent in the data without respect to external information. Two primary metrics of internal validation are cohesion and separation. Cohesion assesses within-cluster variation (i.e. how close objects within the same cluster are based on some distance measures) [92]. A lower value indicates better clustering. Separation, in contrast, evaluates intercluster variation (i.e. how distinct or well separated one cluster is from other clusters based on distances between cluster centers or pairwise distances between objects in different clusters) [92]. A higher value suggests better clustering. More advanced metrics use both cohesion and separation to compute a quality measure. The silhouette index is a popular normalized summation-type metric that simultaneously assesses how close an object is to its cluster (cohesion) and how far it is from objects in the nearest neighboring clusters (separation) [94]. Its value ranges from −1 to 1. The closer to 1, the better the samples are clustered. The Dunn index is the ratio of the minimum intercluster distance to the maximum intracluster distance, with a higher value indicating better clustering [95].

Evaluating the performance of different PCA models means assessing how many PCs should be used to represent the data optimally. The cumulative percentage of the total variation, a widely used metric, is the percentage of the variance accounted for by the first m PCs. A threshold somewhere between 70% and 90% is usually chosen [96]. The Kaiser rule assumes that PCs with eigenvalues greater than unity contain more information than the original variables and should be retained [97]. In practice, a cutoff of 0.7 is recommended [96]. A scree test plots eigenvalues in descending order and identifies the 'elbow' of the graph, where the left components are kept [98]. However, these metrics are subjective, and the threshold is sometimes ambitious [99]. Qin and Dunia [99] proposed the variance of the reconstruction error (VRE) for determining the number of PCs to retain based on the best reconstruction of the variables. The VRE metric guarantees a minimum over the number of PCs to ensure the best reconstruction [99]. Cross-validation is another metric that promises a minimum over the number of PCs based on the estimates, predicted from the model, of some deleted data. The PCs are successively retained until no significant improvement in prediction occurs with the addition of new PCs [96].

Both external and internal metrics can be used to evaluate outlier detection algorithms, although this remains challenging [100]. Common external evaluation metrics are the AUC, precision at *n* (*P*@*n*), average precision (AP), and the adjusted indexes of *P*@*n* and AP [100]. These metrics, however, rely on labeled outliers and inliers [100], which is not practical for real applications. Marques *et al*. [101] have provided the only internal metric so far for assessing the performance of outlier detection algorithms based solely on the data themselves, that is, the Internal, Relative Evaluation of Outlier Solutions (IREOS). IREOS calculates the separability of different sets of candidate outliers on the assumption that an outlier is far away from others and easier to separate [101]. This metric, unfortunately, has been criticized as being computationally expensive [100].

Applications of machine learning algorithms for identifying genes related to plant stress and disease resistance

The temperature and the intensity and frequency of drought are increasing in climate change scenarios, together with the global population. These changes increase humans' need for food [102] and thus for breeding better crops that are tolerant to stress [103]. Machine learning algorithms can help breeders and researchers identify stress resistance genes. Overall, most work has been focused on drought resistance genes. Using a variant of the SVM algorithm, Liang *et al*. [104] identified 10 important genes that may participate in drought resistance in *Arabidopsis thaliana*. Heath *et al*. [105] used inductive logic programming to identify several drought resistance genes that correspond to the production of chaperones and membrane transport proteins in *Pinus taeda*. Shikha *et al*. [106] compared the performance of seven machine learning algorithms, including regularization algorithms, Bayes algorithms, and reproduction of the kernel Hilbert space, to predict breeding values of 240 drought-resistant maize subtropical lines. They found that in general Bayes algorithms performed better than other algorithms. They further identified 77 superior SNPs that were important for drought resistance out of 29 619 curated SNPs in maize [106]. A few studies have used machine learning algorithms to identify candidate genes that are responsive to other stresses. Wang *et al*. [107] developed an SVM-based model for predicting salt resistance genes in rice. Ma *et al*. [17] used a network analysis to identify candidate genes responsive to multiple stresses besides drought and salt, such as heat, cold, wounds, and genotoxicity, in *Arabidopsis thaliana*. They also found that, compared to traditional methods of hypothesis testing (i.e. *t* tests, *F* tests, and analysis of variance), network analysis was better at identifying stress resistance genes [17]. However, these works aside, the current use of machine learning algorithms to identify stress resistance genes is limited to a few species. Extending applications of machine learning to other economically important plants will accelerate both understanding of plant stress resistance mechanisms and breeding.

Identifying plant disease resistance genes (R genes) is another crucial aspect of improving crop production in which machine learning plays an important role. SVM and its variants are the most widely used algorithms for predicting R genes. Pal *et al*. [108] evaluated the performance of SVM for predicting R genes in 25 species. They showed that SVM achieved a high accuracy of 91.11% on a test data set [108], although no other methods were compared. Kushwaha *et al*. [109] found that SVM had good performance for predicting disease resistance proteins in five plant species. Shaik and Ramakrishna [110] compared the performance of recursive-SVM and random forest, PCA, and partial least squares discriminant analysis in differentiating disease resistance genes from stress resistance genes. They found that recursive-SVM and random forest had greater accuracy than the others [110]. Other studies have used machine learning to predict pathogen effector proteins. For example, Sperschneider *et al*. [111] created the first machine learning classifier, EFFECTORP, to predict fungal effectors based on sequence-derived properties. Later they improved the models' accuracy to 89% and recognized four key features for prediction (i.e. protein size, protein net charge, and the amino acids serine and cysteine) [112]. Saunders *et al*. [113] identified eight top-ranked candidate families of rust fungi effectors using hierarchical clustering and Markov clustering. However, a rather important but neglected aspect of research has been the identification of genes susceptible to plant disease [114]. Nevertheless, a broad application of machine learning algorithms will increase understanding of plant–pathogen interactions at the molecular level and contribute to agricultural practice [114].

Conclusion

Machine learning has shown great potential for analyzing enormous highly dimensional data sets, although it has had limited application in plant molecular studies. A better understanding of machine learning algorithms will facilitate the spread of machine learning among plant biologists. Advancements in graphics processing units (GPUs) and the increase in state-of-the-art packages in R provide a platform for easily applying these modern methods. As plant sequencing data continue to accumulate, machine learning will speed up all aspects of plant genomic research, including identifying genes related to resistance to biotic and abiotic stress and other genes with important functions, estimating breeding values, understanding mechanisms of gene regulation and exploring the genetic architecture of genomes. These advancements in turn will assist agricultural researchers in improving the yield and quality of crops with better tolerance to biotic and abiotic stress.

Supplementary data

Supplementary data mentioned in the text are available to subscribers in *BRIFUN* online.

Funding

This work was supported by the National Natural Science Foundation of China (Nos. 91935302, 61771331, 61922020 and 91735306).

References

1. Doring Y, Noels H, Weber C. The use of high-throughput technologies to investigate vascular inflammation and atherosclerosis. *Arterioscler Thromb Vasc Biol* 2012;**32**:182–95.
2. Jiang L, Xiao Y, Ding Y, *et al*. FKL-spa-LapRLS: an accurate method for identifying human microRNA-disease association. *BMC Genomics* 2018;**19**:911.
3. Jiang L, Ding Y, Tang J, *et al*. MDA-SKF: similarity kernel fusion for accurately discovering miRNA-disease association. *Front Genet* 2018;**9**:1–13.
4. Singh A, Ganapathysubramanian B, Singh AK, *et al*. Machine learning for high-throughput stress phenotyping in plants. *Trends Plant Sci* 2016;**21**:110–24.
5. Mitchell TM. *Machine Learning*. New York, USA: McGraw-Hill, 1997.
6. Xu Y, Wang Y, Luo J, *et al*. Deep learning of the splicing (epi) genetic code reveals a novel candidate mechanism linking histone modifications to ESC fate decision. *Nucleic Acids Res* 2017;**45**:12100–12.
7. Zou Q, Mrozek D, Ma Q, *et al*. Scalable data mining algorithms in computational biology and biomedicine. *Biomed Res Int* 2017;**2017**:1–3.
8. Chen W, Lv H, Nie FL, *et al*. i6mA-Pred: identifying DNA N-6-methyladenine sites in the rice genome. *Bioinformatics* 2019;**35**:2796–800.
9. Zhu XJ, Feng CQ, Lai HY, *et al*. Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl-Based Sys* 2019;**163**:787–93.
10. Cheng L, Jiang Y, Ju H, *et al*. InfAcrOnt: calculating cross-ontology term similarities using information flow by a random walk. *BMC Genomics* 2018;**19**:10.
11. Cheng L, Hu Y, Sun J, *et al*. DincRNA: a comprehensive web-based bioinformatics toolkit for exploring disease associations and ncRNA function. *Bioinformatics* 2018;**34**:1953–6.
12. Roy S, Ernst J, Kharchenko PV, *et al*. Identification of functional elements and regulatory circuits by Drosophila modENCODE. *Science* 2010;**330**:1787–97.

13. Lai HY, Zhang ZY, Su ZD, *et al.* iProEP: a computational predictor for predicting promoter. *Mol Ther-Nucleic Acids* 2019;**17**:337–46.
14. Dunham I, Kundaje A, Aldred SF, *et al.* An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;**489**:57–74.
15. Xu YG, Zhao WL, Olson SD, *et al.* Alternative splicing links histone modifications to stem cell fate decision. *Genome Biol* 2018;**19**:21.
16. Tan JX, Lv H, Wang F, *et al.* A survey for predicting enzyme family classes using machine learning methods. *Curr Drug Targets* 2019;**20**:540–50.
17. Ma C, Xin MM, Feldmann KA, *et al.* Machine learning-based differential network analysis: a study of stress-responsive Transcriptomes in Arabidopsis. *Plant Cell* 2014;**26**:520–37.
18. R Core Team (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
19. National Center for Biotechnology Information. The NCBI handbook [Internet]. In: , .
20. Tello-Ruiz MK, Naithani S, Stein JC, *et al.* Gramene 2018: unifying comparative genomics and pathway resources for plant research. *Nucleic Acids Res* 2017;**46**:D1181–9.
21. Goodstein DM, Shu S, Howson R, *et al.* Phytozome: a comparative platform for green plant genomics. *Nucleic Acids Res* 2012;**40**:D1178–86.
22. Osuna-Cruz CM, Paytuvi-Gallart A, Di Donato A, *et al.* PRGdb 3.0: a comprehensive platform for prediction and analysis of plant disease resistance genes. *Nucleic Acids Res* 2017;**46**:D1197–201.
23. Fernandez-Pozo N, Menda N, Edwards JD, *et al.* The Sol Genomics Network (SGN)–from genotype to phenotype to breeding. *Nucleic Acids Res* 2015;**43**:D1036–41.
24. Dash S, Campbell JD, Cannon EKS, *et al.* Legume information system (LegumeInfo.org): a key component of a set of federated data resources for the legume family. *Nucleic Acids Res* 2015;**44**:D1181–8.
25. Berardini TZ, Reiser L, Li D, *et al.* The arabidopsis information resource: making and mining the “gold standard” annotated reference plant genome. *Genesis* 2015;**53**:474–85.
26. Portwood JL, II, Woodhouse MR, Cannon EK, *et al.* MaizeGDB 2018: the maize multi-genome genetics and genomics database. *Nucleic Acids Res* 2018;**47**:D1146–54.
27. The IC4R Project Consortium. Information commons for rice (IC4R). *Nucleic Acids Res* 2016;**44**:D1172–80.
28. Appels R, Eversole K, Stein N, *et al.* Shifting the limits in wheat research and breeding using a fully annotated reference genome. *Science* 2018;**361**:eaar7191.
29. Tan JX, Li SH, Zhang ZM, *et al.* Identification of hormone binding proteins based on machine learning methods. *Math Biosci Eng* 2019;**16**:2466–80.
30. Ding YJ, Tang JJ, Guo F. Identification of drug-target interactions via multiple information integration. *Inform Sci* 2017;**418**:546–60.
31. Liu B, Li C, Yan K. DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. *Brief Bioinform* 2019;bbz098. doi: 10.1093/bib/bbz098.
32. Xu L, Liang GM, Liao CR, *et al.* An efficient classifier for Alzheimer’s disease genes identification. *Molecules* 2018;**23**:13.
33. Lv H, Zhang Z-M, Li S-H, *et al.* Evaluation of different computational methods on 5-methylcytosine sites identification. *Brief Bioinform* 2019;bbz048. doi: 10.1093/bib/bbz048.
34. Ding YJ, Tang JJ, Guo F. Identification of protein-ligand binding sites by sequence information and ensemble classifier. *J Chem Inf Model* 2017;**57**:3149–61.
35. Ding YJ, Tang JJ, Guo F. Predicting protein-protein interactions via multivariate mutual information of protein sequences. *Bmc Bioinformatics* 2016;**17**:13.
36. Liu B, Yang F, Huang DS, *et al.* iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics* 2018;**34**:33–40.
37. Ayers KL, Cordell HJ. SNP selection in genome-wide and candidate gene studies via penalized logistic regression. *Genet Epidemiol* 2010;**34**:879–91.
38. Park MY, Hastie T. Penalized logistic regression for detecting gene interactions. *Biostatistics* 2007;**9**:30–50.
39. Hua S, Sun Z. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics* 2001;**17**:721–8.
40. Zhu PP, Li WC, Zhong ZJ, *et al.* Predicting the subcellular localization of mycobacterial proteins by incorporating the optimal tripeptides into the general form of pseudo amino acid composition. *Mol Biosyst* 2015;**11**:558–63.
41. Shen YN, Tang JJ, Guo F. Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou’s general PseAAC. *J Theor Biol* 2019;**462**:230–9.
42. Zien A, Rätsch G, Mika S, *et al.* Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics* 2000;**16**:799–807.
43. Xu L, Liang GM, Shi SH, *et al.* SeqSVM: a sequence-based support vector machine method for identifying antioxidant proteins. *Int J Mol Sci* 2018;**19**:11.
44. Xu L, Liang GM, Wang LJ, *et al.* A novel hybrid sequence-based model for identifying anticancer peptides. *Gene* 2018;**9**:13.
45. Chen W, Song XH, Lv H, *et al.* iRNA-m2G: identifying N2-methylguanosine sites based on sequence-derived information, molecular therapy. *Nucleic Acids* 2019;**18**:253–8.
46. Yang H, Lv H, Ding H, *et al.* iRNA-2OM: a sequence-based predictor for identifying 2 ’-O-methylation sites in *Homo sapiens*. *J Comput Biol* 2018;**25**:1266–77.
47. Brown MPS, Grundy WN, Lin D, *et al.* Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci* 2000;**97**:262–7.
48. Liu B, Li K. iPromoter-2L2.0: identifying promoters and their types by combining smoothing cutting window algorithm and sequence-based features, molecular therapy. *Nucleic Acids* 2019;**18**:80–7.
49. Dror G, Sorek R, Shamir R. Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics* 2004;**21**:897–901.
50. Goldstein BA, Hubbard AE, Cutler A, *et al.* An application of random forests to a genome-wide association dataset: methodological considerations & new findings. *BMC Genet* 2010;**11**:49.
51. Wu J, Liu H, Duan X, *et al.* Prediction of DNA-binding residues in proteins from amino acid sequences using a random forest model with a hybrid feature. *Bioinformatics* 2009;**25**:30–5.
52. Jiang R, Tang W, Wu X, *et al.* A random forest approach to the detection of epistatic interactions in case-control studies. *Bmc Bioinformatics* 2009;**10**:S65.
53. Xu L, Liang GM, Liao CR, *et al.* K-skip-n-gram-RF: a random Forest based method for Alzheimer’s disease protein identification. *Front Genet* 2019;**10**:7.

54. Ching T, Himmelstein DS, Beaulieu-Jones BK, *et al*. Opportunities and obstacles for deep learning in biology and medicine. *J R Soc Interface* 2018;**15**:1–47.
55. Chen Y, Li Y, Narayan R, *et al*. Gene expression inference with deep learning. *Bioinformatics* 2016;**32**:1832–9.
56. Reese MG. Application of a time-delay neural network to promoter annotation in the Drosophila melanogaster genome. *Comput Chem* 2001;**26**:51–6.
57. Pedersen AG, Nielsen H. Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis, proceedings. International Conference on Intelligent Systems for Molecular Biology. 1997;**5**:226–33.
58. Larsson AJM, Johnsson P, Hagemann-Jensen M, *et al*. Genomic encoding of transcriptional burst kinetics. *Nature* 2019;**565**:251–4.
59. Gorlov IP, Pikielny CW, Frost HR, *et al*. Gene characteristics predicting missense, nonsense and frameshift mutations in tumor samples. *Bmc Bioinformatics* 2018;**19**:430.
60. Friedman J, Hastie T, Tibshirani R. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, USA: Springer, 2001.
61. Bøvelstad HM, Nygård S, Størvold HL, *et al*. Predicting survival from microarray data—a comparative study. *Bioinformatics* 2007;**23**:2080–7.
62. Ogutu JO, Schulz-Streeck T, Piepho H. Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. *BMC Proc* 2012;**6**:S10.
63. Piepho HP. Ridge regression and extensions for genomewide selection in maize. *Crop Sci* 2009;**49**:1165–76.
64. Waldmann P, Mészáros G, Gredler B, *et al*. Evaluation of the lasso and the elastic net in genome-wide association studies. *Front Genet* 2013;**4**:270.
65. Xu D, Tian Y. A comprehensive survey of clustering algorithms. *Ann Data Sci* 2015;**2**:165–93.
66. The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature* 2012;**490**:61–70.
67. D'haeseleer P, Liang S, Somogyi R. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* 2000;**16**:707–26.
68. Wang S, Wong D, Forrest K, *et al*. Characterization of polyploid wheat genomic diversity using a high-density 90 000 single nucleotide polymorphism array. *Plant Biotechnol J* 2014;**12**:787–96.
69. Zhu PF, Xu Q, Hu QH, *et al*. Co-regularized unsupervised feature selection. *Neurocomputing* 2018;**275**:2855–63.
70. Zhu PF, Xu Q, Hu QH, *et al*. Multi-label feature selection with missing labels. *Pattern Recognit* 2018;**74**:488–502.
71. Zhu PF, Zhu WC, Hu QH, *et al*. Subspace clustering guided unsupervised feature selection. *Pattern Recognit* 2017;**66**:364–74.
72. Ding H, Li DM. Identification of mitochondrial proteins of malaria parasite using analysis of variance. *Amino Acids* 2015;**47**:329–33.
73. Liu B, Gao X, Zhang H. BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Res* 2019;**gkz740**:1–12.
74. Hibbs MA, Dirksen NC, Li K, *et al*. Visualization methods for statistical analysis of microarray clusters. *Bmc Bioinformatics* 2005;**6**:115–5.
75. Yeung KY, Ruzzo WL. Principal component analysis for clustering gene expression data. *Bioinformatics* 2001;**17**:763–74.
76. Ma S, Dai Y. Principal component analysis based methods in bioinformatics studies. *Brief Bioinform* 2011;**12**:714–22.
77. Luikart G, England PR, Tallmon D, *et al*. The power and promise of population genomics: from genotyping to genome typing. *Nat Rev Genet* 2003;**4**:981–94.
78. Ghosh D. Discrete nonparametric algorithms for outlier detection with genomic data. *J Biopharm Stat* 2010;**20**:193–208.
79. Liaw A, Wiener M. Classification and regression by randomForest. *R News* 2002;**2**:18–22.
80. Kuhn M. Building predictive models in R using the caret package. *J Stat Softw* 2008;**28**:1–26.
81. Jerome Friedman TH, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Software* 2010;**33**:1–22.
82. Meyer D, Dimitriadou E, Hornik K, *et al*. *e1071: Misc functions of the department of statistics, probability theory group (Formerly: E1071), TU Wien*, 2019.
83. Tiwari V, Kashikar A. OutlierDetection: Outlier Detection. R package version 0.1.1, 2019.
84. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 2009;**21**:1263–84.
85. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. *Inf Process Manag* 2009;**45**:427–37.
86. Feng PM, Chen W, Lin H, *et al*. iHSP-PseRAAAC: identifying the heat shock protein families using pseudo reduced amino acid alphabet composition. *Anal Biochem* 2013;**442**:118–25.
87. Chen W, Feng P, Song X, *et al*. iRNA-m7G: identifying N7-methylguanosine sites by fusing multiple features, molecular therapy. *Nucleic Acids* 2019;**18**:269–74.
88. Liu B. BioSeq-analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Brief Bioinform* 2019;**20**:1280–94.
89. Dao FY, Lv H, Wang F, *et al*. Identify origin of replication in *Saccharomyces cerevisiae* using two-step feature selection technique. *Bioinformatics* 2019;**35**:2075–83.
90. Liu B, Zhu YL. ProtDec-LTR3.0: protein remote homology detection by incorporating profile-based features into learning to rank. *IEEE Access* 2019;**7**:102499–102507.
91. Liu B, Chen SY, Yan K, *et al*. iRO-PsekGCC: identify DNA replication origins based on pseudo k-tuple GC composition. *Front Genet* 2019;**10**:8.
92. Tan P, Steinbach M, Karpatne A, *et al*. *Introduction to Data Mining*. Pearson: New York, USA, 2018.
93. Alexander S, Joydeep G. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J Machine Learning Res* 2002;**3**:583–617.
94. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.
95. Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J Cybernetics* 1973;**3**:32–57.
96. Jolliffe IT. *Principal Component Analysis*. New York, USA: Springer, 2002.
97. Kaiser HF. The application of electronic computers to factor analysis. *Educ Psychol Meas* 1960;**20**:141–51.
98. Cattell RB. The scree test for the number of factors. *Multivar Behav Res* 1966;**1**:245–76.

99. Qin SJ, Dunia R. Determining the number of principal components for best reconstruction. *J Process Control* 2000;**10**:245–50.
100. Campos GO, Zimek A, Sander J, *et al*. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining Knowl Discov* 2016;**30**:891–927.
101. Marques HO, Campello RJGB, Zimek A, *et al*. On the internal evaluation of unsupervised outlier detection. Proceedings of the 27th international conference on scientific and statistical database management. La Jolla. La Jolla, California: ACM. 2015;**7**:1–12.
102. Hasegawa T, Fujimori S, Havlík P, *et al*. Risk of increased food insecurity under stringent global climate change mitigation policy. *Nature Climate Change* 2018;**8**:699–703.
103. Abberton M, Batley J, Bentley A, *et al*. Global agricultural intensification during climate change: a role for genomics. *Plant Biotechnol J* 2016;**14**:1095–8.
104. Liang Y, Zhang F, Wang J, *et al*. Prediction of drought-resistant genes in *Arabidopsis thaliana* using SVM-RFE. *PLoS One* 2011;**6**:e21750.
105. Heath LS, Ramakrishnan N, Sederoff RR, *et al*. Studying the functional genomics of stress responses in loblolly pine with the Expresso microarray experiment management system. *Compar Funct Genomics* 2002;**3**:226–43.
106. Shikha M, Kanika A, Rao AR, *et al*. Genomic selection for drought tolerance using genome-wide SNPs in maize. *Front Plant Sci* 2017;**8**:1–12.
107. Wang J, Chen L, Wang Y, *et al*. A computational systems biology study for understanding salt tolerance mechanism in Rice. *PLoS One* 2013;**8**:e64929.
108. Pal T, Jaiswal V, Chauhan RS. DRPPP: a machine learning based tool for prediction of disease resistance proteins in plants. *Comput Biol Med* 2016;**78**:42–8.
109. Kushwaha SK, Chauhan P, Hedlund K, *et al*. NBSPred: a support vector machine-based high-throughput pipeline for plant resistance protein NBSLRR prediction. *Bioinformatics* 2015;**32**:1223–5.
110. Shaik R, Ramakrishna W. Machine learning approaches distinguish multiple stress conditions using stress-responsive genes and identify candidate genes for broad resistance in rice. *Plant Physiol* 2014;**164**:481–95.
111. Sperschneider J, Gardiner DM, Dodds PN, *et al*. EffectorP: predicting fungal effector proteins from secretomes using machine learning. *New Phytol* 2016;**210**: 743–61.
112. Sperschneider J, Dodds PN, Gardiner DM, *et al*. Improved prediction of fungal effector proteins from secretomes with EffectorP 2.0. *Mol Plant Pathol* 2018;**19**:2094–110.
113. Saunders DGO, Win J, Cano LM, *et al*. Using hierarchical clustering of secreted protein families to classify and rank candidate effectors of rust fungi. *PLoS One* 2012;**7**: e29847.
114. Yang X, Guo T. Machine learning in plant disease research. *Eur J BioMed Res* 2017;**3**:6–9.