Briefings in Bioinformatics, 2025, **26(1)**, bbae593
https://doi.org/10.1093/bib/bbae593
Problem Solving Protocol

AutoXAI4Omics: an automated explainable AI tool for omics and tabular data

James Strudwick[1,‡], Laura-Jayne Gardiner[1,‡], Kate Denning-James[2], Niina Haiminen[3], Ashley Evans[1], Jennifer Kelly[1], Matthew Madgwick[1], Filippo Utro[3], Ed Seabolt[4], Christopher Gibson[1], Bharat Bedi[1], Daniel Clayton[5], Ciaron Howell[5], Laxmi Parida[3], Anna Paola Carrieri[1,*]

[1]IBM Research Europe, The Hartree Centre - Sci-Tech Daresbury, Keckwick Lane, Daresbury, Warrington WA4 4AD, United Kingdom
[2]Earlham Institute, Norwich Research Park, Colney Lane, Norwich NR4 7UZ
[3]IBM T.J. Watson Research Center, 1101 Kitchawan Rd, Yorktown Heights, NY 10598, United States
[4]IBM Research, Almaden, 650 Harry Rd, San Jose, CA 95120, United States
[5]STFC, The Hartree Centre, Sci-Tech Daresbury, Keckwick Lane, Daresbury, Warrington WA4 4AD, United Kingdom

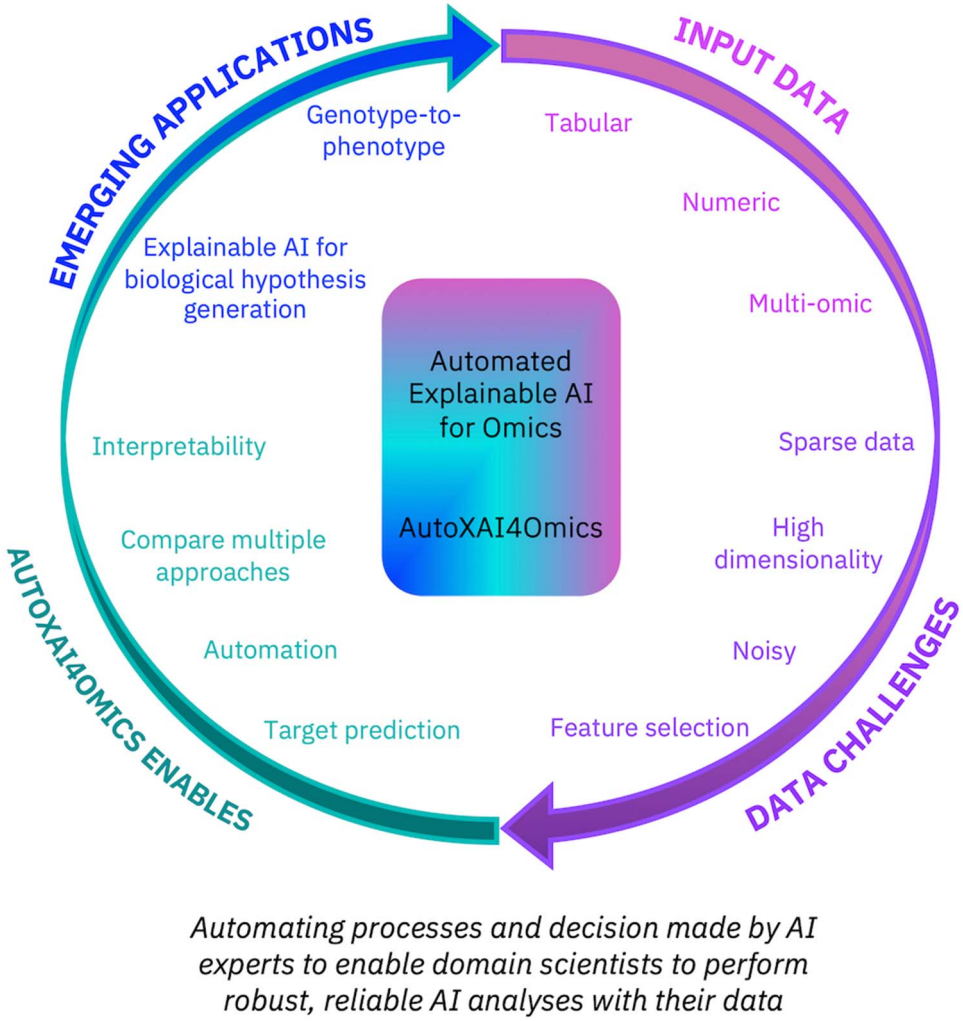
*Corresponding author. E-mail: acarrieri@uk.ibm.com
‡James Strudwick and Laura-Jayne Gardiner authors contributed equally to this work.

Abstract

Machine learning (ML) methods offer opportunities for gaining insights into the intricate workings of complex biological systems, and their applications are increasingly prominent in the analysis of omics data to facilitate tasks, such as the identification of novel biomarkers and predictive modeling of phenotypes. For scientists and domain experts, leveraging user-friendly ML pipelines can be incredibly valuable, enabling them to run sophisticated, robust, and interpretable models without requiring in-depth expertise in coding or algorithmic optimization. By streamlining the process of model development and training, researchers can devote their time and energies to the critical tasks of biological interpretation and validation, thereby maximizing the scientific impact of ML-driven insights. Here, we present an entirely automated open-source explainable AI tool, AutoXAI4Omics, that performs classification and regression tasks from omics and tabular numerical data. AutoXAI4Omics accelerates scientific discovery by automating processes and decisions made by AI experts, e.g. selection of the best feature set, hyper-tuning of different ML algorithms and selection of the best ML model for a specific task and dataset. Prior to ML analysis AutoXAI4Omics incorporates feature filtering options that are tailored to specific omic data types. Moreover, the insights into the predictions that are provided by the tool through explainability analysis highlight associations between omic feature values and the targets under investigation, e.g. predicted phenotypes, facilitating the identification of novel actionable insights. AutoXAI4Omics is available at: https://github.com/IBM/AutoXAI4Omics.

Received: March 27, 2024. **Revised:** September 17, 2024. **Accepted:** November 1, 2024

Graphical Abstract



Keywords: machine learning; automated; explainable; omics

Introduction

In recent years, there has been an increase in utilizing artificial intelligence (AI) and machine learning (ML) within scientific discovery and pipeline development for the analysis of healthcare and life science datasets [1]. AI represents the broader concept of simulating human intelligence by machines, whereas ML is a branch of AI commonly involving a computer learning autonomously. In supervised ML, it is the provision of labels for the input data that enables the learning; algorithms parse the input data (feature sets), learn patterns from it relating to the label (or target), and can then make predictions about the labels for unseen data. ML algorithms are well suited to large, heterogeneous, and complex input data sets, such as omics (e.g. genomics, transcriptomics, proteomics, metabolomics, etc.), that are becoming increasingly common as the cost of sequencing decreases. In response to this, numerous open-source AI and ML tools have been specifically developed for analyzing omics data [2], addressing common analytical questions and challenges associated with large-scale datasets. Research scientists require efficient and automated ML methods to predict outcomes from omics data and to uncover the relationships between omic features and target variables, such as phenotypes. However, despite this need, there is a surprising dearth of entirely automated, end-to-end, open-source ML tools that can handle diverse types of omics data and perform generic classification or regression tasks.

Current state of the art

Focusing on open-source offerings for omics data, there are a wide range of notable AI and ML tools that have been developed for specific use cases, including genotype-to-phenotype prediction (e.g. DeepCOMBI [3], AutoML-GWAS [4], and Emedgene [5]), SNP calling (e.g. SNP-ML [6] and DeepVariant [7]), radiogenomics [8] (e.g. ImaGene [9]), omic feature selection (e.g. [10]), predicting gene regulation (e.g. BioAutoMATED [11], EUGENe [12], and CRMnet [13]), single-cell RNA gene regulatory analysis (e.g. scGeneRAI [14]), circadian rhythm detection (e.g. ZeitZeiger [15]), and protein folding prediction (e.g. Alphafold [16]). Many of these tools present a single ML method and focus on a specific data type or a few selected data types (e.g. genomics and transcriptomics). Furthermore, most of these tools are tailored to a single application

domain, e.g. for biomedical and drug discovery (summarized in the review by Gao et al. [2]) or plant science (summarized in the reviews by Mahood et al. [17] and Gardiner and Krishna [18]).

It is not uncommon for both AI and ML models to be referred to as 'black-boxes', due to their complexity and lack of interpretability. This lack of interpretability (or explainability) can create concern for the user of the model, the domain expert, since it may allow issues and biases to be overlooked [19]. Consequently, explainable AI (XAI) algorithms, such as LIME [20] and SHAP (SHapley Additive exPlanations) [21], are applied to ML models to overcome these problems and improve trust in the models and their results by providing explanations of their predictions [22]. Our previous work has harnessed such interpretability and explainability of models to gain biological insights into the target variable being predicted, e.g. for the prioritization of key genes or markers associated with a predicted phenotype [23–25]. This explainability goes beyond the interpretability offered by the feature importance metrics used in existing workflows since it directly explains the decisions made by a ML model [26, 27].

SHAP is a powerful explainability method that can be applied to any ML or deep learning model, such as neural networks and convolutional neural networks. SHAP is a unifying framework that integrates multiple explainability methods, including LIME, Shapley values, layer-wise relevance propagation, and quantitative input influence feature attributions. By combining game theory with local explanation, SHAP provides accurate interpretations of how the model predicts a particular value for a given sample. The local explanations reveal subtle changes and interrelations that are often missed when differences are averaged out. In addition to providing a ranked list of important features for an ML or deep learning model, SHAP offers several advantages over other feature importance methods. These include the ability to explain how each feature contributes positively or negatively to the prediction of specific phenotypic values. Finally, the problems of 'black-boxes' can be accentuated by the lack of open-source pipelines, i.e. when a user is dependent on a model wrapped in a workflow that is also not transparent.

Our offering: AutoXAI4Omics

In this manuscript, we introduce our open-source, explainable, end-to-end ML tool for omics data and other numerical tabular datasets, AutoXAI4Omics. Figure 1 provides a high-level representation of the AutoXAI4Omics workflow. This software enables users to seamlessly navigate an automated workflow that begins with data ingestion and proceeds through a sequence of critical steps: data pre-processing, such as filtering, normalization, scaling (1-optional), automated feature selection (2-optional), and hyperparameter optimization (3) to generate a suite of optimal ML models tailored to the user's specific task (classification or regression). AutoXAI4Omics then aids the user in the evaluation of these models (4), suggesting the best model (5) based on a range of key considerations (e.g. over-fitting and performance on held-out data) and generating a wide range of visualizations of the results (6). Finally, it facilitates interpretability of results (7), not only via feature importance inference, but using an XAI algorithm to provide the user with a detailed global explanation of the most influential features contributing to each generated model. AutoXAI4Omics can process a large range of tabular data types, including RNA-seq, microbiome-related data, metabolomics, genomics, and other numerical tabular data, e.g. clinical, environmental, and demographic data. These data types can span domains, such as human health and disease, drug discovery, plant phenotyping, and environmental studies. It also has pre-processing capability tailored to omic data, e.g. TMM normalization for transcriptomics and common microbiome sample/feature filtering.

AutoXAI4Omics is available as a containerized command line tool where the intended audience is bioinformaticians or non-expert ML users generally. To use it, the user provides an input tabular data set (samples × features table), metadata (typically the target to predict by the ML), and a configuration file (config, json file). Our workflow builds upon existing open-source ML tools by uniquely integrating an end-to-end workflow, tailored to omics-specific needs, with essential features, such as automated feature selection, best model recommendation, and XAI, prioritizing interpretability of the results. We ensure applicability across diverse data types and domains and provide a fully containerized solution for smooth installation and re-use. AutoXAI4Omics is available on Github at: https://github.com/IBM/AutoXAI4Omics.

In this manuscript, we detail the options available to the user in AutoXAI4Omics, and using three distinct biological case studies, we will demonstrate its simplistic usage, predictive power, and application agnosticism. The various data sets and configuration files that we use here are also provided as supplementary material to encourage ease of re-use, since they can act as a template for new users. Our first biological use case focuses on the plant science domain, where we use AutoXAI4Omics to perform a specific binary classification task. More precisely, we predict if Barley accessions are two-rowed or six-rowed (relating to spikelet fertility) from genomic data. Our second use case is derived from the biomedical domain, where we use human gene expression data to predict cell response to a series of infections (influenza, *Escherichia coli*, and control) as a multi-class classification task. Finally, our third use case focuses on environmental data, where we predict soil pH (a regression task) from normalized soil microbiome species abundance data. For each specific prediction task, we use algorithms including Random Forest (RF), XGBoost (XGB), AdaBoost, K-nearest neighbors (KNN), LightGBM (LGBM), and a neural network (Auto Keras) followed by an XAI algorithm to assess the biological validity of the ML model, i.e. to check if key predictors align with current biological know-how. We show that the XAI component of the workflow can act as an ML model validator and hypothesis generator to rank key omic features that are driving the predictions for a target or phenotype of interest, since these top features are candidates for further analysis or experimental testing.

Results and discussion

Binary classification using AutoXAI4Omics: a case study in plant genomics

ML and high-throughput phenotyping within plant sciences can reduce laborious manual work and cost while improving the efficiency of data collection, e.g. via automated image collection and analysis [28]. ML has also been used to improve the timelines for plant breeding [29, 30]. Omics datasets can be utilized to select genes that underpin traits of interest to enable their incorporation into breeding programs [31]. ML can fine-tune the selection of SNPs or QTLs from genomic datasets within genome-wide association studies and genomic selection. New ML tools are being continually developed for understanding traits of interest in plants, for investigating gene expression [17] and for integrating multi-omic datasets to predict phenotypes [32, 33].

We downloaded a genotyping matrix of global GBS (genotype-by-sequencing) information plus related phenotype data for 1000 core Barley (*Hordeum vulgare L.*) accessions published in [34]. In

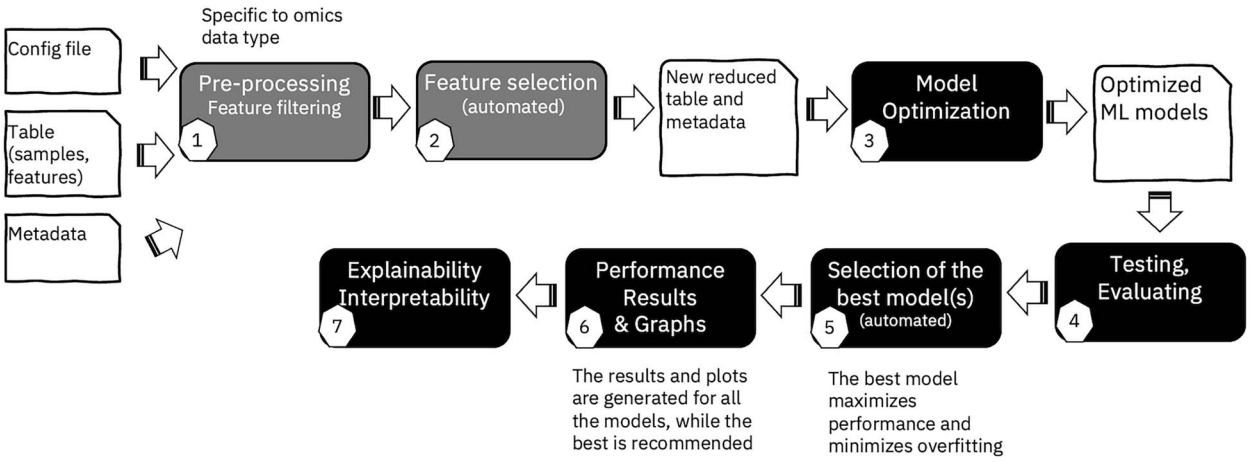


Figure 1. Overview of the AutoXAI4Omics XAI workflow from data input to results and interpretation.

this data set, Barley accessions are categorized as two-rowed or six-rowed based on their lateral spikelets and floret sizes. Six-rowed Barley has three spikelets, all of which are fertile and can produce grain. While two-rowed Barley has three spikelets, but only the central spikelet is fertile [35]. We converted phenotypic data into binary format: two-rowed (0) or six-rowed (1) and converted the features or SNPs into numerical format for usage in AutoXAI4Omics; homozygous for reference allele (2), heterozygous (1), homozygous for alternative allele (0), and NNs (3). The data set thus included 957 Barley accessions (after removing those with intermediate phenotypes or missing data) and 37 953 features (SNPs), which were used to train ML models to perform classification for row-number.

AutoXAI4Omics was run for this data set using classification mode, a random search, f1-scoring and a train:test split ratio of 80:20. The config file that was used to run this analysis is provided as File S1.json and the associated data/metadata files to run this config as File S2.csv and File S3.csv, respectively. Figure 2a shows the performance of a range of hyper-tuned ML models for the prediction of the target during cross validation. Several models perform well, though no two are identical. Our encoded algorithm within AutoXAI4Omics suggests the 'best' or 'recommended' model for the user to focus on, XGBoost. Elements of this decision are visible in Fig. 2a, where XGBoost is generating a high accuracy coupled to a low performance variation across the different folds on cross-validation. What is not shown in Fig. 2a is additional information (considered by AutoXAI4Omics) regarding minimal overfitting observed when comparing training and test data performance (f1-score) and balanced performance across the predicted classes (i.e. performance was most balanced for XGBoost). Figure 2b depicts the confusion matrix, and Fig. 2c presents the ROC curve for the recommended model, XGBoost, with F1-scores of 1.0 on the training data and 0.98 on the test data (these values and others are reported as a results table for the user in Table S1). Finally, Fig. 2d shows the accuracy curve using an increasing number of selected input features using a random forest classifier.

Figure 3 summarizes the XAI results as computed by SHAP algorithm for the Barley binary classification highlighting the top 15 features and their relative importance's for the classification via SHAP. The y-axis, in Fig. 3a–c, displays the results of the XAI analysis, which ranks the relative importance or SHAP impact of the input features (SNPs represented as chromosome:position) on the best ML model (XGBoost)—features are ranked from high impact to low impact from top to bottom of the y-axis. Figure 3a shows a bar plot summarizing the top 15 feature values for the best model (XGBoost) with their absolute average impact on the model predictive output; Fig. 3b and c are summary dot plots displaying the top 15 features (y-axis) against their SHAP impact values (x-axis). Rows are features, dots are samples, and the color of a dot represents the feature value (SNPs encoded as 0, 1, or 2) of that sample for the corresponding feature. A positive impact value of a feature for a sample represents a positive association with that feature value and the sample being assigned to a specific class, either six-rowed (Fig. 3b) or two-rowed Barley Fig. 3c. Figure 3 allows us to interpret the results and validate the ML by linking predictive features to biological processes. For example, the two most predictive SNPs of row number in Barley from AutoXAI4Omics are within chromosome 2 at position 2:651372029 and 2:651378029 in order of their relative importance. We searched for genes within a 20 kb region around these SNPs utilizing GrainGenes [36]. Two high-confidence genes were found within this region based on the Barley cv. Morex V3 annotation [37]; HORVU.MOREX.r3.2HG0211880 (Exostosin family protein); and HORVU.MOREX.r3.2HG0211900 (Phytol kinase 1). HORVU.MOREX.r3.2HG0211900 has been associated with the differentiation of two-rowed and six-rowed Barley in prior research [38], utilizing Barlex [39]. This gene may also be linked to the loci *six-rowed spike vrs*, *hexastichon (hex-v)*, or *intermedium spike (int)*, which are heavily discussed in the literature due to their association with row number in Barley [35, 40–43]. Further evidence to support that the genes identified are linked to row number and spike development in Barley is from expression data, which suggests their expression during the seed development and within spike meristems [44, 45]. Finally, orthologues in wheat (*Triticum aestivum*) with over a 90% sequence match to the genes from Barley were found using ensembl [46]. The orthologues were then investigated further utilizing KnetMiner [47], which associated them to pathways within root and flower meristems and for seed development and germination.

Multi-class classification using AutoXAI4Omics: a case study for human RNA-seq

We downloaded and used the Series Matrix text file for study GSE53165 that is part of the NCBI Gene Expression Omnibus. This data set was generated as part of a prior publication [48] and is related to the variation in individuals' responses to complex diseases in humans. The authors generated transcriptional

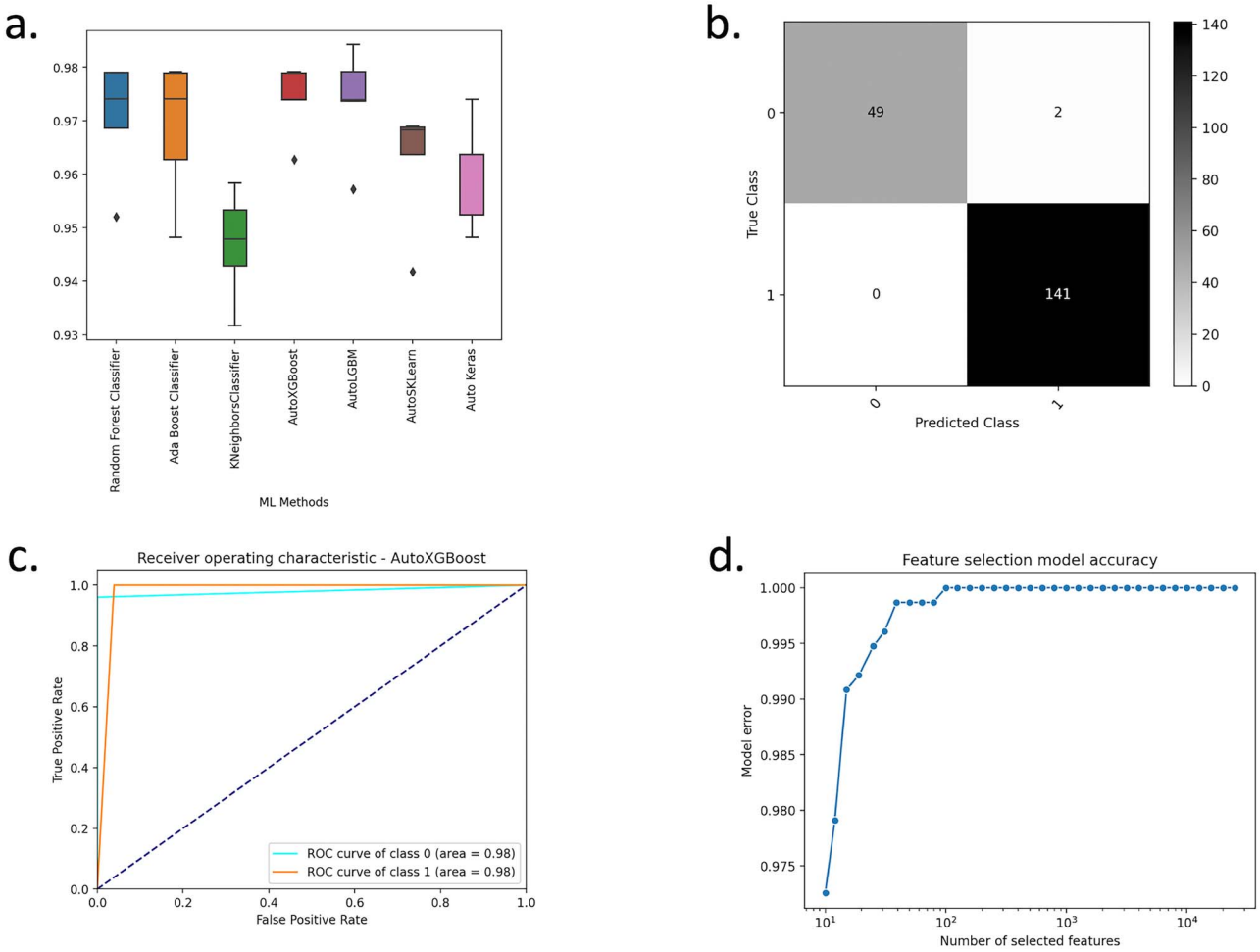


Figure 2. Binary classification using AutoXAI4Omics: a case study in plant genomics. Figure summarises the predictive performance of AutoXAI4Omics in predicting either two-rowed (0) or six-rowed barley (1). (a) Box plots displaying the f1-score during cross validation, (b) confusion matrix for the best performing ML model (XGBoost), (c) ROC curve for the best performing model, and (d) feature selection accuracy curve.

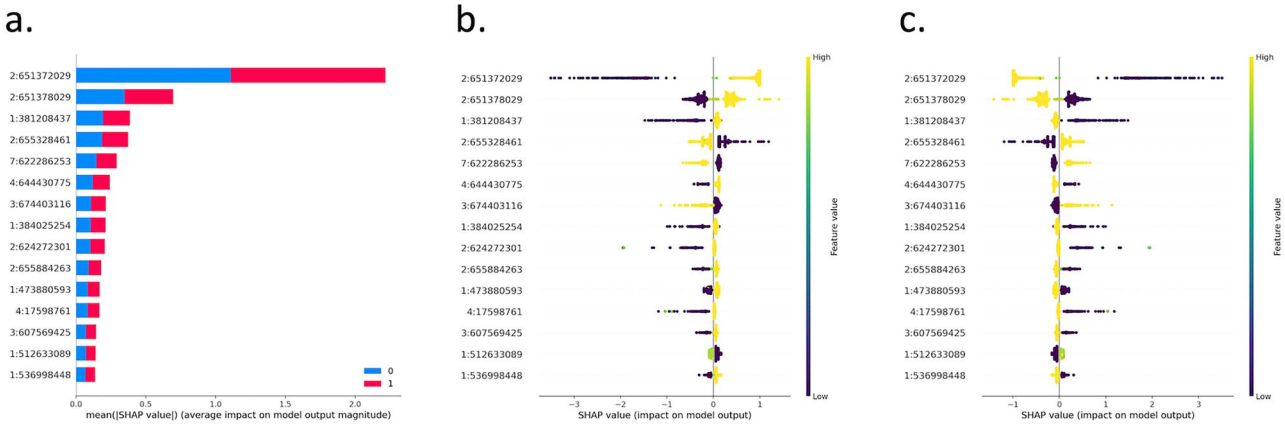


Figure 3. Binary classification using AutoXAI4Omics: a case study in plant genomics. Figure summarizes the XAI output relating to the best performing model from AutoXAI4Omics(XGBoost) to predict either two-rowed (0) or six-rowed barley (1). (a) Bar plot summarizing the top 15 feature values selected for the best model (XGBoost), (b) SHAP global view of explanations for six-rowed predictions, and (c) SHAP global view of explanations for two-rowed Barley predictions.

profiles from dendritic cells (DCs), derived from the peripheral blood of healthy individuals that were subjected to different states. Here we compare three of these states: resting (unstim), *E. coli* lipopolysaccharide stimulated (LPS), and influenza-infected (dNS1). We used this data set, encompassing 2009 samples (that we reduced to 167 by removing serial replicates) and 414 features (genes with expression information), to train several ML models implemented in AutoXAI4Omics to perform multi-class classification. The targets to predict were the three treatments: unstim, LPS, and dNS1. The config file used to run this analysis

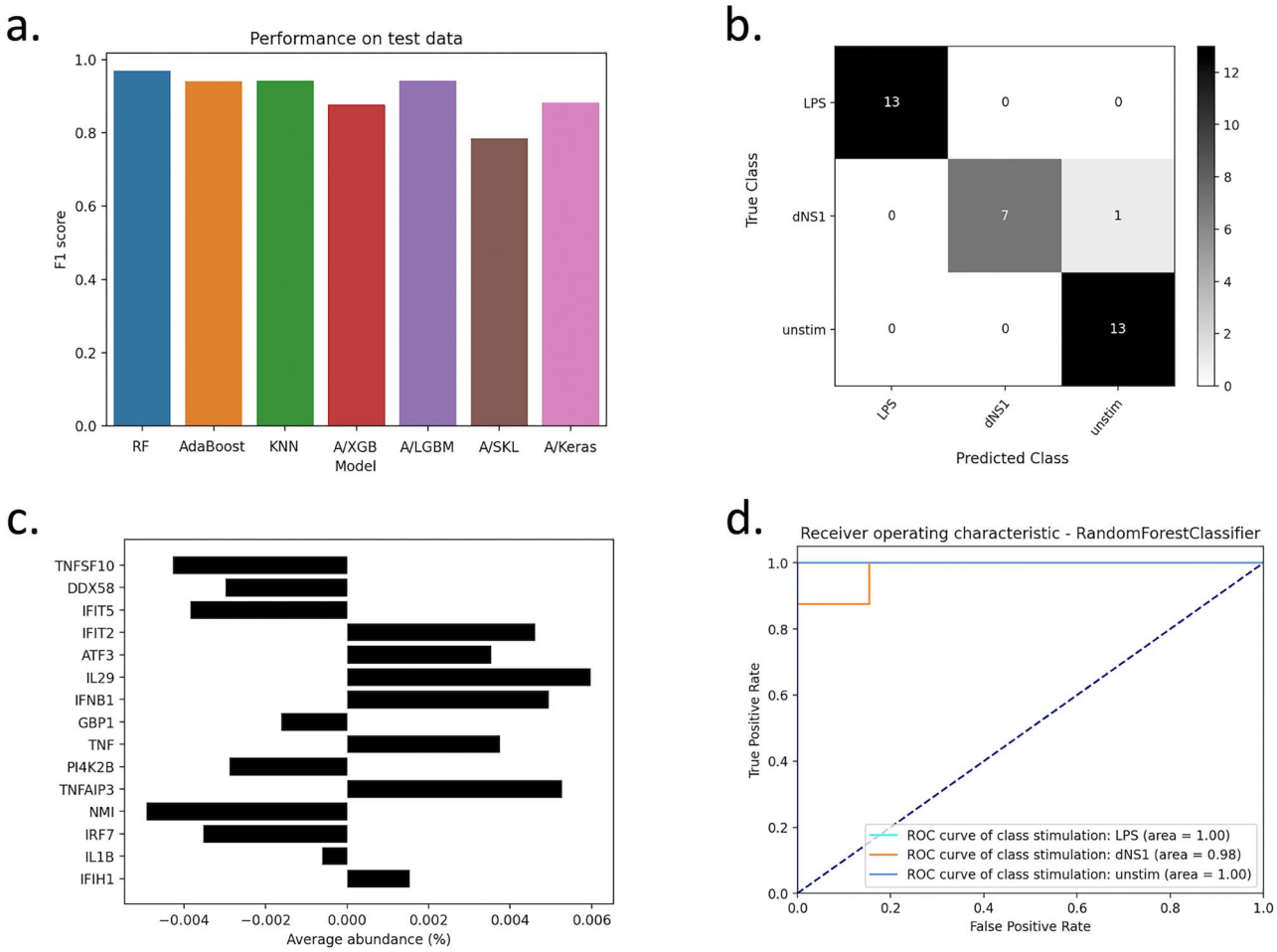


Figure 4. Multi-class classification using AutoXAI4Omics: a case study with human RNA-seq data. (a) Bar charts displaying the f1-score on the held-out test dataset, (b) confusion matrix for the best performing ML model (random forest), (c) bar chart displays feature rank from feature importance analysis with their average abundance, and (d) ROC curve for the best performing ML model.

in AutoXAI4Omics is attached as File S4.json and the associated data/metadata files to run this config as File S5.csv and File S6.csv, respectively. Most notably for this analysis, we used classification mode, a random search, f1-scoring, automated feature selection, and automated filtering for samples more than 5 SD away from the mean and those with adjusted gene expression counts of 1 or less in 10 or less samples. Figure 4 summarizes the performance of the AutoXAI4Omics supervised ML workflow to predict three classes: unstim, LPS, and dNS1 from human transcriptomic data. Figure 4a shows the performance across all models (test data), and Fig. 4b–d focuses on our best performing model as identified by AutoXAI4Omics (Random Forest), where the confusion matrix (Fig. 4b) and ROC curves (Fig. 4d) highlight robust performance across all classes. The weighted F1-scores across the three classes for our best model were 0.99 on the training data and 0.97 on the held-out test data (Table S2).

Figure 4c ranks the relative importance of the input features prior to ML model development (y-axis); the x-axis depicts the average abundance of the related features across the input sample set (average gene expression counts). Figure 4 gives insight into the relative importance's of the input features to the model, but the advantage of our XAI workflow is a more comprehensive interpretation of the generated models via SHAP. Figure 5 shows the SHAP output for our best performing ML model (Random Forest) for each of the three classes: unstim, LPS, and dNS1. We often see features having a mirror image role in different classes (as per binary classification), e.g. for the most predictive gene (dNS1 class) Interleukin-29 (IL29), a higher abundance positively associates with a sample being linked to the dNS1 class, while a lower abundance positively associates with a sample being linked to the LPS class. Furthermore, we advocate using these XAI outputs to sanity check the biological validity of the ML model, i.e. to check if key predictors align with current biological know-how. In support of this, the most predictive gene for class dNS1 (also the fourth most predictive for class LPS) was IL29. IL29 is a cytokine that has been linked to antiviral activity, antibacterial activity, antiproliferative activity, and *in vivo* antitumour activity [49], and higher IL29 expression has been linked to influenza infection, i.e. the dNS1 class (which is the pattern reflected in Fig. 5b). Conversely, for LPS induction of IL29, previous work noted high expression of IL29 in DCs after initial LPS stimulation (0–2 h) and then a steep drop-off in IL29 expression after further LPS exposure (2–18 h) [50]; since the cells used in our ML analysis were exposed to LPS for more than 2 h (up to 5 h) this aligns with the low-mid expression of IL29 driving a prediction of class LPS (seen in Fig. 5a). Fittingly IL29 is not present in the most predictive features for the unstim class where no infective agent is present.

Many other predictive genes align with common sense as to what might be relevant for response to disease or stress, e.g. Activating transcription factor 3 (ATF3) a stress-induced transcription

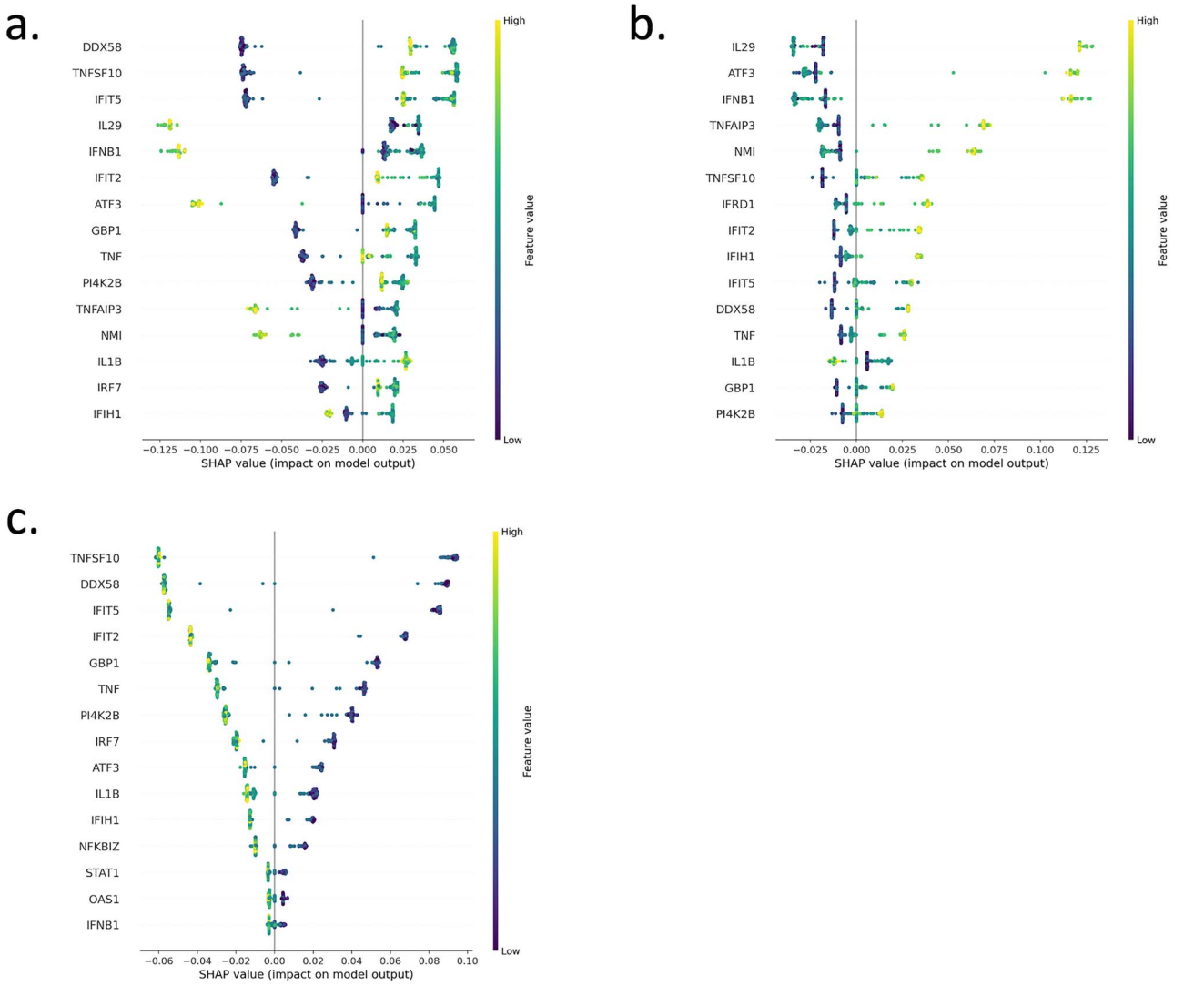


Figure 5. Multi-class classification using AutoXAI4Omics: a case study with human RNA-seq data. XAI output relating to the best ML model (random forest) to predict three classes unstim, LPS, and dNS1. (a) Global explanation for class LPS, (b) global explanation for class dNS1, and (c) global explanation for class unstim.

factor, where high levels of this gene make a sample more likely to be assigned to the dNS1 class, mid-expression levels of the gene are associated with the LPS class, and low levels with the unstim class. Also, guanylate-binding protein 1 (GBP1) plays important roles in innate immunity against a diverse range of bacterial, viral, and protozoan pathogens and follows broadly the same predictive profile as ATF3, i.e. high levels make a sample more likely to be assigned to the dNS1 class, mid-expression levels are associated with the LPS class, and low levels with the unstim class.

Regression using AutoXAI4Omics: a case study for environmental microbiome data

We used a bioinformatics workflow (see Methods) to generate species abundance information for 189 environmental microbiome samples that were part of a study presented by Bahram *et al.* [51] comprising whole shotgun metagenomes from samples of topsoil collected from representative terrestrial regions and biomes across the world. Based on sample geo-locations, we gathered information for each sample regarding the pH of the soil at the sampled depth (0–5 cm) from SoilGrids [52]. We used this data set, encompassing 189 samples (that we reduced to 173 by removing low-quality samples) and 91 837 features (species with normalized abundance information), to train a series of ML models to predict pH as a regression task. The config file used to run this analysis in AutoXAI4Omics is attached as File S7.json and the associated data/metadata files to run this config as File S8.csv and File S9.csv, respectively. To run this analysis, most notably, we used regression mode, a grid search, mean absolute error (MAE) scoring, automated feature selection, and automated filtering for samples more than 5 SD deviations away from the mean and those with abundances of 5 or less in 10 or less samples.

Figure 6 summarizes the results from the AutoXAI4Omics analysis for the prediction of soil pH from soil microbiome data. Figure 6a is a bar plot showing the predictive performance, as MAE, across all models on the held-out test data. Figure 6b shows MAE on cross validation for each model, and Fig. 6c–d focuses on our best performing model as identified by AutoXAI4Omics (Random Forest), where the correlation plot (Fig. 6c) and joint plot (Fig. 6d) highlight the high degree of correlation between true and predicted values. The MAEs for our best model were 0.02 on the training data and 0.31 on the held-out test data. AutoXAI4Omics outputs additional evaluation metrics, and we found the metrics

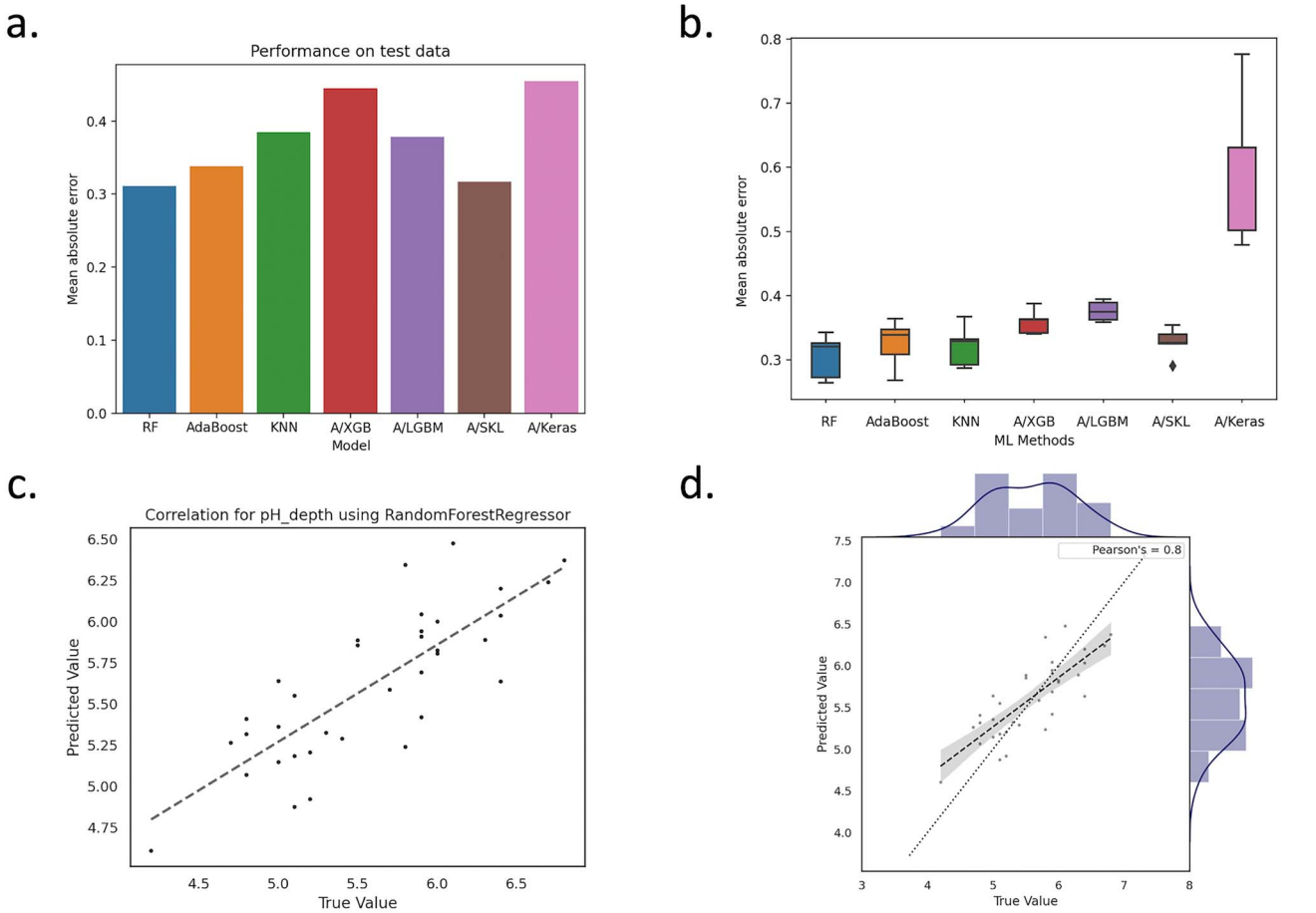


Figure 6. Regression using AutoXAI4Omics: a case study with environmental microbiome data. Predictive performance of AutoXAI4Omics to predict soil pH from the soil microbiome. (a) Bar charts displaying the MAE on the held-out test data set across a range of ML models, (b) box plot displaying MAE on cross validation across a range of models, (c) correlation, and (d) joint plot to compare predicted values (y-axis) with true values (x-axis) for the best performing ML model (random forest). Diagonal dotted line is also shown.

mean absolute percentage error (MAPE) and R2 can also be useful to assess performance. For this analysis, there was an MAPE of only 5.7% on the held-out test data and a relatively high R2 of 0.64.

Figure 7 shows the SHAP output of the top 15 most predictive species for our best-performing ML model (Random Forest) for soil pH. A global explanation is shown where a low feature value (blue) indicates a low species abundance, and a higher feature value (red) indicates a higher species abundance for a sample represented as a dot. If a higher feature value correlates with a higher SHAP value (the sample/dot is on right-hand side of the plot axis), then a higher species abundance tends to predict a higher pH value. Conversely, if a higher feature value correlates with a lower SHAP value (the sample/dot is on the left-hand side of the plot axis), then a higher species abundance tends to predict a lower pH value. Some full names have been shortened as follows: *Gammaproteobacteria bacterium RIFCSPHIGHO2 12 FULL 45 9* is shortened to *Gammaproteobacteria bacterium RIF 12 45 9* and *Acidobacteria bacterium RIFCSPHIGHO2 02 FULL 67 57* shortened to *Acidobacteria bacterium* RIF 02 67 57. Notably, the most predictive species include several acidophiles or acid-linked species, e.g. *Acidocella* sp. *MX-AZ02*, *Ralstonia* virus RSL1, and an *Acidobacteria* bacterium. As one might expect, these species show a positive association with the prediction of a low pH, i.e. higher species abundances tend to associate with lower or negative SHAP values that lower the model's predicted pH for a sample. Additionally, the most predictive species, *Candidatus nitrosotalea* sp. FS has known associations with acidic soil that fit both its appearance as a predictor and also the directionality of its effect (i.e. higher abundance predictive of a lower pH). Moreover, species within its associated genus, *Ca. nitrosotalea*, are reportedly ammonia oxidizers abundant in acidic soil environments and obligate acidophiles, growing optimally around pH 5.0, which directly supports our observed importance and trends [53].

Comparison with state of art

With the aim of explicitly distinguishing AutoXAI4Omics from existing alternatives, in the Section 'Current State of Art', we referenced a wide range of notable AI and ML tools developed for specific omics use cases (e.g. genotype-to-phenotype prediction, SNP calling, predicting gene regulation, single-cell RNA gene regulatory analysis, and protein folding prediction). However, these tools differ significantly from AutoXAI4Omics in terms of their functionalities, data types, and biological questions.

Most of these tools are designed to address a single application domain or focus on a specific use case. Their input and output formats also differ from those required by AutoXAI4Omics. An example is GenoML [54], which is restricted to binary classification (0/1) and regression from SNPs inputted in standard PLINK files, without support for multi-class classification or interpretability/-explainability methods. Some of these fundamental differences in

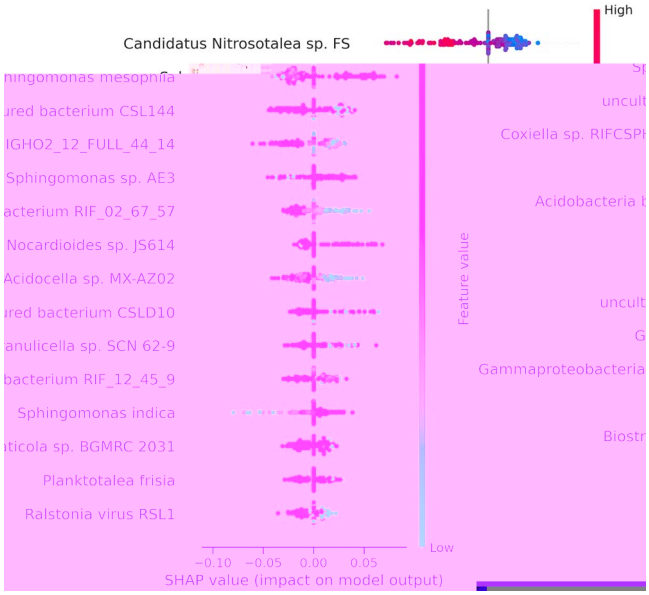


Figure 7. Regression using AutoXAI4Omics: a case study with human RNA-seq data. Figure summarizes the XAI output from SHAP relating to the best ML model generated by AutoXAI4Omics (random forest) to predict soil pH.

design and purpose render it impossible or invalid to numerically compare these tools with AutoXAI4Omics.

The most similar tool to AutoXAI4Omics that we identified is OmicLearn [55], an ML pipeline specifically designed for proteomics data but also applicable to other types of omics data presented in tabular format. To further emphasize the unique features and capabilities of AutoXAI4Omics, in Table S3, we summarize the key differences when comparing OmicLearn to AutoXAI4Omics. A critical difference lies in their prediction capabilities: while AutoXAI4Omics supports binary, multi-class, and regression tasks, OmicLearn is limited to binary classification only. As demonstrated by our three use cases in plant genomics, human transcriptomics, and environmental microbiome samples, this limitation restricts the applicability of OmicLearn for a broader range of applications. In contrast, AutoXAI4Omics incorporates several key features that are not implemented in OmicLearn: omics-specific pre-processing to handle complex data, class balancing techniques (e.g. oversampling) to improve model performance, automated feature selection to identify the most informative variables, hyper-parameter tuning for optimal modeling capabilities, automated selection of the best performing model for a given task, and interpretability methods, such as SHAP and eli5, that enable researchers to understand the relationships between features and predictions. We believe that the comprehensive set of features in AutoXAI4Omics makes it a more versatile tool for omics data analysis.

In terms of numerical comparisons, we evaluated predictive performance on our first use case (binary classification of two-rowed versus six-rowed plants from SNPs) using both tools. Our dataset consisted of 956 samples and 37 953 features. Notably, the free deployment of OmicLearn (https://ol-v14.streamlit.app/) was unable to handle this data size, resulting in a server crash. When we downloaded the local dashboard, we were able to run the analysis, but it failed to return the full list of performance metrics as described in their manual, either in image or CSV formats. The only visualization provided was a ROC curve with an AUC of 0.98 computed by XGBoost (Fig. S1), which matched our best model's AUC (0.98) using AutoXAI4Omics (Fig. 2c).

Conclusions

In this study, we present our open-source end-to-end XAI tool, AutoXAI4Omics. We apply it to three use cases, and we provide a series of examples (including related data and configuration files) to showcase the utility and performance of the tool. Our use cases cover a wide range of omic data types (genomics, transcriptomics, and microbiome), application domains (plant science, environment, and human disease), tasks (multi-class/binary classification and regression), and pre-processing options (automated feature selection, user-defined feature selection, and feature/sample filtering). We also focus on the advantage offered by the inclusion of XAI approaches alongside feature importance techniques. In doing so, we highlight the wide applicability of our tool to a wide range of research areas to help scientists answer a variety of life sciences questions. AutoXAI4Omics can be used by domain scientists that are not ML experts to quickly perform robust and trustworthy interpretable ML analysis and therefore focus on the insights generated by the tool, rather than in building and fine-tuning ML models, hence accelerating scientific discovery.

Methods

AutoXAI4Omics workflow

AutoXAI4Omics has four modes that can be invoked, which are:

- *Training:* train, tune, and evaluate several ML algorithms to perform classification or regression tasks on a given dataset as specified in the configuration file.
- *Plotting:* create (or re-create) plots and visualizations for a trained and optimized model or a set of models as specified in the config.
- *Holdout:* apply the trained and tuned model(s) to a holdout separate dataset and evaluate their predictive performance on it.
- *Prediction:* makes predictions on new, unlabeled data, using the best-trained model and gets its explanations.

The flow that we describe below includes those steps that all belong to and run in the training mode. The other modes invoke the corresponding stages that are required for their individual tasks, e.g. plotting only or predicting only on a holdout dataset or new unlabeled data points.

Load data and *data specific processing*

The first step AutoXAI4Omics performs is to load the data into the system. At this point, if the user chooses, omic-specific pre-processing is performed on the data. This is an optional step and is only done if the user sets the corresponding field in the configuration file, data_type. We enable this step to be optional as there are many ways that this pre-processing can be done, with users possibly having their own preferred method. If a user has already pre-processed their data, then they may delete the omic-specific entry from the config, and this will stop any pre-processing from happening. Also, AutoXAI4Omics can accept and work with non-omic data, if a user wishes, and as before, the user only needs to remove the omic-specific pre-processing field from the config. AutoXAI4Omics includes pre-processing modules for gene expression data, microbiome data, metabolomic, and tabular numerical data.

The gene expression pre-processing is designed to accept RNA-seq read counts and microarray expression levels and is activated when the user specifies in the config data type = 'gene expression'. The user needs to also specify the type of their expression data, e.g. 'FPKM', 'RPKM', 'TMM', 'TPM', 'Log2FC', 'COUNTS', 'OTHER', where 'COUNTS' invokes conversion of raw count data to normalized trimmed mean of M-values (TMM) values. Other additional pre-processing capabilities that are popular for those using gene expression data have been added to allow filtering of unwanted or potentially problematic genes or samples. First, the function 'filter sample' removes samples if the number of genes with coverage is more than X standard deviations from the mean across all samples, i.e. the extremes of very sparse or high coverage samples. Second, the function 'filter genes' removes genes unless they have a gene expression value over X in Y or more samples, i.e. removing genes that never appear with sufficient expression across the sample and may be noise.

The omic-specific pre-processing also includes several functions specifically for microbiome data. These functions are available when the user specifies data type = 'microbiome' in the configuration file. Microbiome-specific pre-processing capabilities include optional collapsing of taxonomy to kingdom, 'k'; phylum, 'p'; class, 'c'; order, 'o'; family, 'f'; genus, 'g'; or species, 's' levels, which is invoked by the user specifying the taxonomic rank letter in the 'collapse tax' option in the configuration file. Two read-based filtering options are available; 'min reads', which will remove samples with fewer reads than specified, and 'norm reads', which will rescale all samples to sum the number of reads specified. Both 'min reads' and 'norm reads' have default values of 1000, which will be invoked if data type = 'microbiome' is selected and the user does not change the default values in the configuration file. Taxon-based filtering is available with functions 'filter abundance', which will remove taxonomic features with a total count less than the specified number across all samples, and 'filter prevalence', which will remove taxonomic features which occur in less than the specified proportion of samples. The default value for 'filter abundance' is 10. For 'filter prevalence' the default value is 0.01, which represents the decimal percentage of the samples, so using the default value here, any taxonomic features present in <1% of samples would be removed. Metadata-based filtering is also supported for microbiome data. The function 'filter microbiome samples' allows for the removal of samples based on metadata categories. This function requires a dictionary (or list of dictionaries), where the key represents the metadata column (i.e. 'COUNTRY') and the value represents the metadata category in that column to be removed (i.e. 'UK'). The default value for this function is 'null', invoking no metadata-based filtering. Two final functions are available that can manipulate the 'target' metadata category specifically. These include 'remove classes', which takes in a list of class labels to be removed from the 'target' column, and 'merge classes', which allows for merging of target classes by taking in a dictionary in the format '{"X": ["A"," B"]}', which would convert class labels 'A' and 'B' to 'X'. Both 'remove classes' and 'merge classes' have default values of 'null', invoking no manipulation of the 'target' column.

The omic-specific pre-processing also supports data types 'tabular' and 'metabolomic' and these have associated filtering options, similar to the gene expression data, for (i) removing samples if the number of measures with measurements is more than X std from the mean across all samples, and (ii) for removing measures unless they have a value over X in Y or more samples.

AutoXAI4Omics only loads and pre-processes one data type per run. For now, if a user has a data set that has non-omic and omic features, then they can do any pre-processing first, outside of the tool themselves, and then give it to AutoXAI4Omics as a 'non-omic' data set. This ability to integrate multiple and different omic types automatically is a feature that we plan to bring to the tool in the future. One other feature that we plan to add later is the automated encoding of any categorical features. For now if the user has any categorical features, then they must manually convert them into one-hot encoded columns first.

Split data

Once the previous step has been completed, the data is split into train-test sets as characterized by the setting in the config that has been presented by the user. If not presented, then the testing size will default to 20%. The exact implementation is done by scikit learns' *train test split* [56]. The split will depend on the random seed that has been set, which also impacts the initial model weights and performance. This can be set in the config provided by the user, but if not provided, it will be set to a default value.

Standardize data

Next the tool standardizes the columns of the training data, for which we use scikit-learns.

QuantileTransformer [56]: ideally when training AI models data should be normally distributed, but in real-world situations, this is rarely the case, and even standard normalization is not completely immune to skewed datasets. This is the reason why we choose to use the *QuantileTransformer* as it is more robust to skewed data.

Feature selection (optional)

Feature selection then follows this step and is completely optional; if the user does not wish feature selection to take place, then they need only remove the corresponding entry from the config. However, if the user does wish to perform feature selection on the training data, then there are several options that they can customize/set. The first step that occurs is variance thresholding (*variance removal* [56]), to remove constant or near-constant columns. The value for this can be set by the user, but if not provided, it will default to 0. Once this is done, the tool can either search for a specified number of best features or it can automatically find the best number of features.

In the case of finding a specified number of best features, then the tool will find them using the setting given by the user; if not present, it will default to using *SelectKBest* method with either the *f classif* or *f regression* metric as implemented by scikit learn [56]. *SelectKBest* works by taking the scoring metric provided to it and selecting the K best scoring features. If using *f classif*, this is the ANOVA *F*-value of the column in relation to the targets, and *f regression* works by running univariate linear regression tests with respect to the target.

If the user then chooses to let the tool automatically find the best number of features, they can choose to set the minimum and maximum number of features the tool will search between. For efficiency, the tool creates several potential candidates that are logarithmically spaced between these two values. Then for each candidate number, a set of best feature features is found, and a default model is trained and evaluated on it. Once all candidates have been scored, we then use our own unique method to select the best candidate number of features to proceed through to the next step. To select the value for *K*, we look for a set of three consecutive values that are the best performing and the most stable; once we have that, then we choose the largest k in that

set as the candidate to use. A stable set of three values has a low standard deviation, and we factor this in is to make sure that our selection is robust. Hypothetically, if an optimum K has terribly performing neighbors, making a 'V' shape, then this k should be treated with an abundance of caution as the addition or removal of a feature causes a drastic chance and could have been achieved because of any number of reasons.

Class balancing (optional, problem specific)

Class balancing is only available when performing classification problems and is also optional. If a user does not want to implement, then they just need to set the corresponding variable in the config. If, however, they do wish to perform class balancing, then there is the option to perform both over-sampling and under-sampling, implemented by imblearn [57].

Train and evaluate models

Now the training of the models can begin. Models available to train include those from scikit learn [54], Auto-scikitlearn [58], xgboost [59], lightgbm [60], and autokeras [61]. These can be trained with most standard metrics that are available, such as Accuracy, F1 score, and Precision for classification tasks; Mean squared error, MAPE, and R^2 for regression tasks. In addition, these models can also be hypertuned on the training data with random or grid search, or without if so desired. A full explanation of all the possible combinations can be found in the code repository. The exact models that are trained are controlled by a corresponding entry in the config the user provides, along with what metric they want to fit according to and what hyper-tuning to use.

Plotting

Once training is complete, if the user desires, plots can be produced illustrating the model performance and explaining how the models work. These plots are optional, and the exact ones produced are controlled by a corresponding list in the users config, meaning it can be skipped if set to empty.

But if plots are desired, these are produced using a combination of seaborn [62], matplotlib [63], and scikit learn [56]. The full range of plots is described in our code repository, but examples are bar and box plots of model performances plus ROC curves and confusion matrices if performing a classification task. Many of these are shown in the Results and Discussion section of this paper.

Explainability

For the explainability aspect, the two libraries we utilize are eli5, for permutation importance, and SHAP, for SHAP value bar and dot plots—again examples of these can be seen in Figs. 3–7. The user can also define and control the number of features that are calculated/plotted via the config file and also specify what data the SHAP values are to be calculated based on, i.e. the training set, test set, or both. In addition to generating SHAP and permutation importance plots, AutoXAI4Omics saves Shapley values and permutation importance values in samples x features tables that return as csv files (permutimp_Top Features_info_[NameOfModel].csv and shapley_values_all_[Name OfModelAndClassIfApplicable].csv, respectively).

A Shapley value is a positive or negative contribution of a feature for a given sample prediction. The Shapley values table enables the end-user to explore these local explanations in more detail and potentially create additional visualizations. The local explanation generated by SHAP provides a nuanced understanding of the relationships between features and predictions.

Select best model

The last step in the process is to select the best model out of those that were trained during the run of AutoXAI4Omics. This selection is done with respect to the metric that the user sets for the models to optimize. For both problem types, classification and regression, the training error and test error are collected and plotted. In the regression problem, the point that is closest to the y = x and the origin (0,0) is selected as the best model. For classification problems, it is the same, but we switch the origin for the unit point (1,1) as in classification problems, the higher performance value the better and values are bounded above by 1. Sometimes this may result in ties, in which case AutoXAI4Omics will pick one but make a point of noting this in their own text file. Once the best model is selected, all of the related content is copied to a dedicated folder, 'best model'.

Datasets

In the Results and Discussion section, we showcase running AutoXAI4Omics with three types of data, including genomics, transcriptomics, and microbiome abundance data. For transcriptomic data this tends to be numerical, and so by nature is compatible with most ML algorithms, but we recommend using normalizations to optimize between sample comparisons, e.g. such as TMM, that we include in AutoXAI4Omics pre-processing. This numerical nature also applies to microbiome data, where users can input normalized or rarefied counts or relative abundance data. Here, we also recommend testing centered log-ratio transformation that removes compositional artifacts. In contrast, genomic data, such as SNPs, can commonly be encountered as a series of alleles; e.g. for diploid organisms, we might see homozygous reference allele calls, heterozygous calls, and homozygous alternate allele calls. For such data, we recommend converting these calls to numerical representations. We have found increasing representative numbers from 0–2 as the presence of an alternate allele increases, i.e. moving from homozygous reference (0) to heterozygous (1) and then homozygous alternate calls (2), can be an intuitive way to represent this type of data, allowing more meaningful interpretation downstream, e.g. to assess as the alternate allele increases what is the effect on the predicted target variable.

Preparation of our three example types of data (genomic, transcriptomic, and microbiome) for usage with AutoXAI4Omics is detailed in the main text. For the genomic and transcriptomic data processed, data was available from public sources that we state. However, we generated the microbiome abundance data from raw sequencing reads that were available as part of the study by Bahram *et al.* [51] at the European Bioinformatics Institute—Sequence Read Archive database under PRJEB24121. We used a bioinformatic analysis workflow for the taxonomic annotation of paired end reads. In our workflow, using Trimmomatic v2.9 [64], reads were trimmed of adapter, and if the average quality across a 4 bp window was less than 15, reads below 40 bp were dropped from the analysis. Reads were then used as input into DIAMOND v2.0.11 [65] for read alignment in blastx mode using '-b 25 -k 5 –index-chunks 4 –min-score 50 and –max-target-seqs 2' options. For these DIAMOND aligned reads, the suite of tools from the MEGAN community edition v6.21.12 [66] were used to run a last common ancestor analysis to allow binning of read pairs by taxon. MEGAN outputs were then processed to extract and compute species-level abundance (read counts).

Key Points

- AutoXAI4Omics: an easy-to-use, open-source, end-to-end, command-line, automated, and XAI tool for omics and tabular data.
- The novelty of AutoXAI4Omics lies in its seamless integration of ML capabilities, streamlined decision-making processes, and intuitive feature analysis. Specifically, it offers the following unique advantages:
  - Classification or regression tasks from omics data in tabular format using state-of-the-art ML algorithms, including neural networks, without requiring extensive coding expertise an algorithmic optimization knowledge.
  - Automated decision making by AI experts, encompassing features such as:
  - + Feature selection to identify the most informative omics variables
  - + Model optimization for optimal performance
  - + Selection of the best-performing model from a suite of options
  - Comprehensive explainability analysis that links omics features to an endpoint, providing actionable insights into the underlying biology.
  - Optional pre-processing capabilities are stailored to specific omics data types, ensuring high-quality input for accurate and reliable results.
- We showcase the versatility and predictive capability of our general-purpose tool, AutoXAI4Omics, through three exemplary use cases, which collectively demonstrate its ability to tackle specific prediction tasks by presenting three use cases that cover different omic data types (genomics, transcriptomics, microbiome) and application domains (plant science, environment, and human disease).

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Author contributions

All authors contributed code and/or conceptualization of AutoXAI4Omics functionality. L.J.G., K.D.-J., A.P.C., and J.S. performed manuscript writing, A.P.C. and L.J.G. performed manuscript review and editing.

Conflict of interest: The authors declare that they have no competing interests. Commercial affiliations do not alter adherence of the authors to the journal policies on sharing data and materials.

Funding

This work was supported by the Hartree National Centre for Digital Innovation (HNCDI), a collaboration between STFC and IBM. K.D.-J. PhD was supported by UKRI-BBSRC through the Norwich Research Park Doctoral Training programme (#2578607).

Data availability

The experimental datasets that were used in this study are all publicly available and details of how to access them can be found in the main text at the first mention of the dataset. The processed datasets used directly in AutoXAI4Omics are provided with this manuscript as Files S1–S9.

Consent for publication

All authors have consented to publication of the manuscript. No additional consent required.

References

1. Wang H, Fu T, du Y. *et al.* Scientific discovery in the age of artificial intelligence. *Nature* 2023;**620**:47–60. https://doi.org/10.1038/s41586-023-06221-2.
2. Gao F, Huang K, Xing Y. Artificial intelligence in omics. *Genomics Proteomics Bioinformatics* 2023;**20**:811–3. https://doi.org/10.1016/j.gpb.2023.01.002.
3. Mieth B, Rozier A, Rodriguez JA. *et al.* DeepCOMBI: explainable artificial intelligence for the analysis and discovery in genomewide association studies. *NAR Genomics and Bioinformatics* 2021;**3**:lqab065. https://doi.org/10.1093/nargab/lqab065.
4. Lakiotaki K, Papadovasilakis Z, Lagani V. *et al.* Automated machine learning for genome wide association studies. *Bioinformatics* 2023;**39**:btad545. https://doi.org/10.1093/bioinformatics/btad545.
5. Ibanez K, Polke J, Hagelstrom RT. *et al.* Whole genome sequencing for the diagnosis of neurological repeat expansion disorders in the UK: a retrospective diagnostic accuracy and prospective clinical validation study. *Lancet Neurol* 2022;**21**:234–45. https://doi.org/10.1016/S1474-4422(21)00462-2.
6. Korani W, Clevenger JP, Chu Y. *et al.* Machine learning as an effective method for identifying true single nucleotide polymorphisms in Polyploid plants. *Plant Genome* 2019;**12**:180023. https://doi.org/10.3835/plantgenome2018.05.0023.
7. Poplin R, Chang PC, Alexander D. *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nat Biotechnol* 2018;**36**:983–7. https://doi.org/10.1038/nbt.4235.
8. Liu Z, Duan T, Zhang Y. *et al.* Radiogenomics: a key component of precision cancer medicine. *Br J Cancer* 2023;**129**:741–53. https://doi.org/10.1038/s41416-023-02317-8.
9. Sukhadia SS, Tyagi A, Venkataraman V. *et al.* ImaGene: a web-based software platform for tumor radiogenomic evaluation and reporting. *Bioinformatics Advances* 2022;**2**:vbac079. https://doi.org/10.1093/bioadv/vbac079.
10. Adornetto C, Greco G. A new deep learning and XAI-based algorithm for features selection in genomics. *arXiv* 2023. 2303.16914[q-bio.GN].
11. Valeri JA, Soenksen LR, Collins KM. *et al.* BioAutoMATED: an end-to-end automated machine learning tool for explanation and design of biological sequences. *Cell Syst* 2023;**14**:525–542.e9. https://doi.org/10.1016/j.cels.2023.05.007; https://www.sciencedirect.com/science/article/pii/ S2405471223001515.
12. Klie A, Laub D, Talwar JV. *et al.* Predictive analyses of regulatory sequences with EUGENe. *Nat Comput Sci* 2023;**3**:946–56. https://doi.org/10.1038/s43588-023-00544-w.
13. Ding K, Dixit G, Parker BJ. *et al.* CRMnet: a deep learning model for predicting gene expression from large regulatory sequence datasets. *Front Big Data* 2023;**6**:e1113402. https://doi.org/10.3389/fdata.2023.1113402.

14. Keyl P, Bischoff P, Dernbach G. *et al.* Single-cell gene regulatory network prediction by explainable AI. *Nucleic Acids Res* 2023;**51**:e20. https://doi.org/10.1093/nar/gkac1212.
15. Hughey JJ, Hastie T, Butte AJ. ZeitZeiger: supervised learning for high-dimensional data from an oscillatory system. *Nucleic Acids Res* 2016;**44**:e80. https://doi.org/10.1093/nar/gkw030.
16. Jumper J, Evans R, Pritzel A. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* 2021;**596**:583–9. https://doi.org/10.1038/s41586-021-03819-2.
17. Mahood EH, Kruse LH, Moghe GD. Machine learning: a powerful tool for gene function prediction in plants. *Appl Plant Sci* 2020;**8**:e11376. https://doi.org/10.1002/aps3.11376.
18. Gardiner L-J, Krishna R. Bluster or lustre: can AI improve crops and plant health? *Plan Theory* 2021;**10**:2707. https://doi.org/10.3390/plants10122707.
19. Bodria F, Giannotti F, Guidotti R. *et al.* Benchmarking and survey of explanation methods for black box models. *Data Min Knowl Disc* 2023;**37**:1719–78. https://doi.org/10.1007/s10618-023-00933-9.
20. Ribeiro MT, Singh S, Guestrin C. Why should I trust you?': Explaining the predictions of any classifier. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101. San Diego, California: Association for Computational Linguistics.
21. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, pp. 4768–77. Red Hook, NY, USA: Curran Associates Inc.
22. Ali S, Abuhmed T, el-Sappagh S. *et al.* Explainable artificial intelligence (XAI): what we know and what is left to attain trustworthy artificial intelligence. *Inf Fusion* 2023;**99**:101805. https://doi.org/10.1016/j.inffus.2023.101805.
23. Carrieri AP, Haiminen N, Maudsley-Barton S. *et al.* Explainable AI reveals changes in skin microbiome composition linked to phenotypic differences. *Sci Rep* 2021;**11**:4565. https://doi.org/10.1038/s41598-021-83922-6.
24. Gardiner LJ, Rusholme-Pilcher R, Colmer J. *et al.* Interpreting machine learning models to investigate circadian regulation and facilitate exploration of clock function. *Proc Natl Acad Sci* 2021;**118**:e2103070118. https://doi.org/10.1073/pnas.2103070118.
25. Gardiner L-J, Carrieri AP, Bingham K. *et al.* Combining explainable machine learning, demographic and multi-omic data to inform precision medicine strategies for inflammatory bowel disease. *PloS One* 2022;**17**:e0263248. https://doi.org/10.1371/journal.pone.0263248.
26. Leclercq M, Vittrant B, Martin-Magniette ML. *et al.* Large-scale automatic feature selection for biomarker discovery in high-dimensional OMICs data. *Front Genet* 2019;**10**:00452. https://doi.org/10.3389/fgene.2019.00452.
27. Bouirdene S, Leclercq M, Quitté L. *et al.* BioDiscViz: a visualization support and consensus signature selector for BioDiscML results. *PloS One* 2023;**18**:e0294750. https://doi.org/10.1371/journal.pone.0294750.
28. Xu R, Li C. A review of high-throughput field phenotyping systems: focusing on ground robots. *Plant Phenomics* 2022;**2022**:9760269. https://doi.org/10.34133/2022/9760269.
29. van Dijk ADJ, Kootstra G, Kruijer W. *et al.* Machine learning in plant science and plant breeding. *iScience* 2021;**24**:101890. https://doi.org/10.1016/j.isci.2020.101890.
30. Cembrowska-Lech D, Krzemińska A, Miller T. *et al.* An integrated multi-omics and artificial intelligence framework for advance plant phenotyping in horticulture. *Biology* 2023;**12**:1298 https://doi.org/10.3390/biology12101298.
31. Najafabadi MY, Hesami M, Eskandari M. Machine learning-assisted approaches in modernized plant breeding programs. *Genes* 2023;**14**:777. https://doi.org/10.3390/genes14040777.
32. Wang K, Abid MA, Rasheed A. *et al.* DNNGP, a deep neural network-based method for genomic prediction using multi-omics data in plants. *Mol Plant* 2023 Celebrating 15 Years of Publication Special Issue;**16**:279–93issn: 1674-2052. https://doi.org/10.1016/j.molp.2022.11.004.
33. Zhang R, Zhang C, Yu C. *et al.* Integration of multi-omics technologies for crop improvement: status and prospects. *Front Bioinform* 2022;**2**:1027457. https://doi.org/10.3389/fbinf.2022.1027457.
34. Bayer MM, Rapazote-Flores P, Ganal M. *et al.* Development and evaluation of a barley 50k iSelect SNP Array. *Front Plant Sci* 2017;**8**:01792. https://doi.org/10.3389/fpls.2017.01792.
35. Nadolska-Orczyk A, Rajchel IK, Orczyk W. *et al.* Major genes determining yield-related traits in wheat and barley. *Theor Appl Genet* 2017;**130**:1081–98. https://doi.org/10.1007/s00122-017-2880-x.
36. Yao E, Blake VC, Cooper L. *et al.* GrainGenes: a data-rich repository for small grains genetics and genomics. *Database* 2022;**2022**:baac034. https://doi.org/10.1093/database/baac034.
37. Mascher M, Wicker T, Jenkins J. *et al.* Long-read sequence assembly: a technical evaluation in barley. *Plant Cell* 2021;**33**:1888–906issn: 1040-4651. https://doi.org/10.1093/plcell/koab077.
38. Wang J-H, Xu Z-M, Qiu X-B. *et al.* Genetic and molecular characterization of determinant of six-rowed spike of barley carrying vrs1.a4. *Theor Appl Genet* 2021;**134**:3225–36. https://doi.org/10.1007/s00122-021-03887-y.
39. Colmsee C, Beier S, Himmelbach A. *et al.* BARLEX – the barley draft genome explorer. *Mol Plant* 2015;**8**:964–6. https://doi.org/10.1016/j.molp.2015.03.009.
40. Guo Y, Himmelbach A, Weiss E. *et al.* Six-rowed wild-growing barleys are hybrids of diverse origins. *Plant J* 2022;**111**:849–58. https://doi.org/10.1111/tpj.15861.
41. Liller CB, Neuhaus R, von Korff M. *et al.* Mutations in barley row type genes have pleiotropic effects on shoot branching. *PloS One* 2015;**10**:e0140246. https://doi.org/10.1371/journal.pone.0140246.
42. Lundqvist U. Scandinavian mutation research in barley – a historical review. *Hereditas* 2014;**151**:123–31. https://doi.org/10.1111/hrd2.00077.
43. Koppolu R, Anwar N, Sakuma S. *et al.* Six-rowed spike4 (Vrs4) controls spikelet determinacy and row-type in barley. *Proc Nat Acad Sci USA* 2013;**110**:13198–203. https://doi.org/10.1073/pnas.1221950110.
44. Waese J, Fan J, Pasha A. *et al.* ePlant: visualizing and exploring multiple levels of data for hypothesis generation in plant biology. *Plant Cell* 2017;**29**:1806–21. https://doi.org/10.1105/tpc.17.00073.
45. Thiel J, Koppolu R, Trautewig C. *et al.* Transcriptional landscapes of floral meristems in barley. *Sci Adv* 2021;**7**:eabf0832. https://doi.org/10.1126/sciadv.abf0832.
46. Martin FJ, Amode MR, Aneja A. *et al.* Ensembl 2023. *Nucleic Acids Res* 2022;**51**:D933–41. https://doi.org/10.1093/nar/gkac958.
47. Hassani-Pak K, Singh A, Brandizi M. *et al.* KnetMiner: a comprehensive approach for supporting evidence-based gene discovery and complex trait analysis across species. *Plant Biotechnol J* 2021;**19**:1670–8 https://knetminer.org. https://doi.org/10.1111/pbi.13583.

48. Lee MN, Ye C, Villani AC. *et al.* Common genetic variants modulate pathogen-sensing responses in human dendritic cells. *Science* 2014;**343**:1246980. https://doi.org/10.1126/science.1246980.
49. Uze G, Monneron D. IL-28 and IL-29: newcomers to the interferon family. *Biochimie* 2007;**89**:729–34. https://doi.org/10.1016/j.biochi.2007.01.008.
50. Williamson MA. The role of Il-29 and Il-28b in the innate immune response. 2018;**1**:0–49.
51. Bahram M, Hildebrand F, Forslund SK. *et al.* Structure and function of the global topsoil microbiome. *Nature* 2018;**560**:233–7. https://doi.org/10.1038/s41586-018-0386-6.
52. Poggio L, de Sousa LM, Batjes NH. *et al.* SoilGrids 2.0: producing soil information for the globe with quantified spatial uncertainty. *Soil* 2021;**7**:217–40. https://doi.org/10.5194/soil-7-217-2021.
53. Prosser JI, Nicol GW. Candidatus Nitrosotaleales†. In: Trujillo ME, Dedysh S, DeVos P, Hedlund B, Kämpfer P, Rainey FA, Whitman WB (eds.), *Bergey's Manual of Systematics of Archaea and Bacteria*. 2016. https://doi.org/10.1002/9781118960608.gbm01292.
54. Makarious M, Leonard HL, Vitale D. *et al.* GenoML: automated machine learning for genomics. *arXiv, preprint arXiv*2103.03221 2021;1–7. https://doi.org/10.48550/arXiv.2103.03221.
55. Torun FM, Virreira Winter S, Doll S. *et al.* Transparent exploration of machine learning for biomarker discovery from proteomics and omics data. *J Proteome Res* 2022;**22**:359–67. https://doi.org/10.1021/acs.jproteome.2c00473.
56. Pedregosa F, Varoquaux G, Gramfort A. *et al.* Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;**12**:2825–30.
57. Lemaˆıtre G, Nogueira F, Aridas CK. Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res* 2017;**18**:1–5. http://jmlr.org/papers/v18/16-365.html.
58. Feurer M, Eggensperger K, Falkner S. *et al.* Auto-sklearn 2.0: hands-free AutoML via meta-learning. *J Mach Learn Res* 2022;**23**:261, 61.
59. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pp. 785–94. New York, NY, USA: Association for Computing Machinery, 2016. https://doi.org/10.1145/2939672.2939785.
60. Zhang H, Si S, Hsieh C-J. GPU-acceleration for large-scale tree boosting. *arXiv* 2017. 1706. 08359[stat.ML].
61. Jin H, Chollet FC, Song Q. *et al.* AutoKeras: an AutoML library for deep learning. *J Mach Learn Res* 2023;**24**:1–6. http://jmlr.org/papers/v24/20-1355.html.
62. Waskom ML. Seaborn: statistical data visualization. *J Open Source Softw* 2021;**6**:3021. https://doi.org/10.21105/joss.03021.
63. Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng* 2007;**9**:90–5. https://doi.org/10.1109/MCSE.2007.55.
64. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 2014;**30**:2114–20. https://doi.org/10.1093/bioinformatics/btu170.
65. Buchfink B, Reuter K, Drost HG. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat Methods* 2021;**18**:366–8. https://doi.org/10.1038/s41592-021-01101-x.
66. Huson DH, Beier S, Flade I. *et al.* MEGAN Community edition – interactive exploration and analysis of large-scale microbiome sequencing data. *PLoS Comput Biol* 2016;**12**:e1004957. https://doi.org/10.1371/journal.pcbi.1004957.

Briefings in Bioinformatics, 2025, **26(1)**, bbae593
https://doi.org/10.1093/bib/bbae593
Problem Solving Protocol