W228–W236 *Nucleic Acids Research, 2021, Vol. 49, Web Server issue* *Published online 25 May 2021*
https://doi.org/10.1093/nar/gkab407

G2PDeep: a web-based deep-learning framework for quantitative phenotype prediction and discovery of genomic markers

Shuai Zeng [1,2], Ziting Mao[1,2], Yijie Ren[1,2], Duolin Wang[1,2], Dong Xu [1,2,3] and Trupti Joshi [1,2,3,4,*]

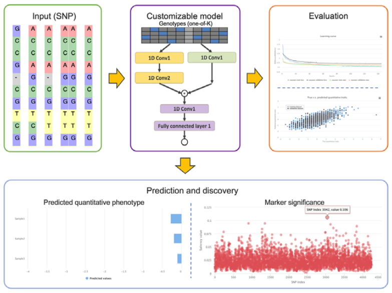
[1]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA, [2]Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO 65211, USA, [3]MU Institute for Data Science and Informatics, University of Missouri, Columbia, MO 65211, USA and [4]Department of Health Management and Informatics, University of Missouri, Columbia, MO 65211, USA

Received March 02, 2021; Revised April 28, 2021; Editorial Decision April 30, 2021; Accepted May 03, 2021

ABSTRACT

G2PDeep is an open-access web server, which provides a deep-learning framework for quantitative phenotype prediction and discovery of genomics markers. It uses zygosity or single nucleotide polymorphism (SNP) information from plants and animals as the input to predict quantitative phenotype of interest and genomic markers associated with phenotype. It provides a one-stop-shop platform for researchers to create deep-learning models through an interactive web interface and train these models with uploaded data, using high-performance computing resources plugged at the backend. G2PDeep also provides a series of informative interfaces to monitor the training process and compare the performance among the trained models. The trained models can then be deployed automatically. The quantitative phenotype and genomic markers are predicted using a user-selected trained model and the results are visualized. Our state-of-the-art model has been benchmarked and demonstrated competitive performance in quantitative phenotype predictions by other researchers. In addition, the server integrates the soybean nested association mapping (SoyNAM) dataset with five phenotypes, including grain yield, height, moisture, oil, and protein. A publicly available dataset for seed protein and oil content has also been integrated into the server. The G2PDeep server is publicly available at http://g2pdeep.org. The Python-based deep-learning model is available at https://github.com/shuaizengMU/G2PDeep_model.

GRAPHICAL ABSTRACT



INTRODUCTION

Genomic selection (GS), one type of marker-assisted selection (MAS) strategy, originally proposed by Meuwissen (1) for animal breeding, has made significant improvement in quantitative phenotype prediction for breeding. It utilizes single nucleotide polymorphisms (SNP) to predict quantitative phenotypes and enhancing traits in breeding populations. Identifying SNP markers allows researchers to increase the effectiveness of identifying candidate genes that affect the diversities of phenotypes, such as protein and oil content in soybean.

With the advances in next-generation sequencing (NGS) technologies, large amounts of SNP data have been generated and are publicly available. Recently, many GS applications have been developed and widely used in bioinformatics studies. These applications provide a fast and low-cost approach comparing to lengthy experimental methods. The rrBLUP (2) is an R package to estimate phenotype by ridge regression with a relationship matrix and Gaussian kernel. DeepGS (3), another R package, applies a deep convolutional neural network and fully connected neural network to predict phenotype from genotypes. Several GS tools based on Bayesian ridge regression have been launched for

[*]To whom correspondence should be addressed. Tel: +1 573 884 5963; Email: joshitr@health.missouri.edu

crops, such as soybeans, rice, corn and oil palm, as well as domesticated animals such as Holstein dairy cattle (4–7).

Despite the availability of these applications and databases, there is still no web-based service available to provide phenotype prediction and genomic marker discovery. The off-line applications have steep learning curves and require complicated installations. Typically, these applications do not provide an easy-to-use interface to create and train a complex deep-learning model efficiently. These applications require users to spend extensive time on the pipeline implementation, dataset creation and transformation into an appropriate model, performance summarization among trained models, and visualization of predicted quantitative phenotype and genomic markers. Furthermore, due to the large size of the SNP datasets for phenotype prediction and marker discovery, these applications often require intensive computing resources that exceed the standard desktop machines, especially for those demanding more memory for model training purposes. Therefore, users must access high-performance computing Linux resources and get familiar with running analyses in such environments, which is a daunting task for many users accustomed to using less technical interfaces.

To address this issue, we have developed G2PDeep, an open-access web server providing a deep-learning framework for quantitative phenotype prediction and genomic marker discovery. The model deployed in G2PDeep was introduced by Liu *et al.* (8) in 2019. It uses the dual-CNN layer, which contains two parallel CNN streams (9), extracting features from one-hot binary coding of genotypes. It also uses a fully connected neural network to predict quantitative phenotype. The saliency map (10), first introduced for visualization of image features in classification, is applied to evaluate the contribution of genomic markers to the phenotype of interest. The method has been benchmarked by other predictors and has always ranked in the first or second place. Both the model and the saliency map are deployed in the webserver. To the best of our knowledge, G2PDeep is the first web-based deep-learning framework available for quantitative phenotype prediction and genomic marker discovery. Unlike other extant related applications, G2PDeep provides an interactive interface enabling deep-learning models to be created, trained and monitored live with dashboards for model performance and reporting, which is unique in the field of bioinformatics. Trained models are stored in G2PDeep and can be retrieved easily by users, making the models reusable and reproducible. G2PDeep applies the guest privacy policy enabling users to retrieve their datasets, models and results without logging. G2PDeep provides real-time predictions for a large number of genotype data using CPU resources, and visualization for predicted phenotype and markers associated with the corresponding phenotype. Currently, G2PDeep integrates the soybean nested association mapping (SoyNAM) (11) dataset, as well as five phenotypes including grain yield, height, moisture, oil, and protein in soybean. A publicly available SNP dataset provided by Bandillo *et al.* (12) in 2015 for seed protein and oil content with over 12 000 unique *G. max* accessions is also available on the server. G2PDeep supports two types of genotype data, i.e. zygosity (homozygous, heterozygous and reference homozygous) and SNP from plants and animals. G2PDeep provides an easy-to-use web interface for genomic selection studies and integrates several publicly available datasets, thus, serving as a valuable tool for the breeders and research community.

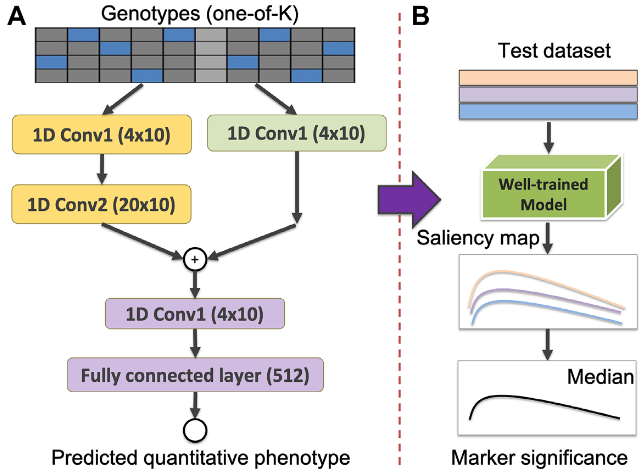


Figure 1. (**A**) Architecture of dual-stream CNN model. The genotypes are one-hot coded. The layers in the left stream are two CNN layers with kernel sizes of 4 and 20, respectively, and the same number of filters 10. The layer in the right stream is a single CNN layer with a kernel size of 4 and number of filters 10. The add-up layer aggregates the output from two streams, followed by a single CNN layer with a kernel size of 4 and number of filters 10. The fully connected layers with numbers of neurons 512 and 1 are regression blocks to predict quantitative phenotype. (**B**) Flowchart of genomic markers discovery using a well-trained model and saliency map. The test dataset is used to estimate the marker significance. For each sample in the test dataset, the saliency map and well-trained model are used to estimate saliency values. The marker significance is calculated by the mean saliency value for each marker position.

MATERIALS AND METHODS

Deep-learning architecture

The architecture of the default model in G2PDeep for quantitative phenotype prediction is shown in Figure 1A. We treated the quantitative phenotype prediction problem as a regression problem. The model takes zygosity and SNP data as input. For zygosity data, three genotypes (homozygous, heterozygous, reference homozygous) and missing data are encoded using one-hot binary encoding. For SNP data, four genotypes (adenine (A), thymine (T), cytosine (C) and guanine (G)) and missing data are also encoded by one-hot binary encoding. Each genomic marker is represented by a four-dimensional and a five-dimensional vector, for zygosity and SNP data respectively, with 1 for a corresponding genotype and 0 for the rest of the genotypes. The output of the model is a numeric value representing the quantitative phenotype. The model consists of a dual-CNN layer, and a fully connected neural network. The encoded genotypes are passed into dual-CNN, which contains two parallel CNN streams, followed by the single-CNN to enhance the representation of markers. The flattened layer integrated representation of markers is passed to a fully connected neural network with a single neuron in the output layer to estimate the quantitative phenotype. The detailed description of the architecture is described in Supplementary Text S1.

Marker significance estimation using saliency map

G2PDeep utilizes the saliency map to investigate and quantify marker significance to a specific phenotype. As shown in Figure 1B, given a set of genotype data provided by users, the quantitative phenotype and marker significance are estimated by the user-selected well-trained model. For each sample, the values of saliency range from 0 to 1, representing markers lowly and highly associated with a single phenotype, respectively. The final marker significance is determined by the median saliency on a whole set of data.

Web server implementation

G2PDeep is applied with a mature and convention-over-configuration Model-View-Controller (MVC) framework, maintained on the CyVerse advanced computing infrastructure (13,14) and hosted on Docker (https://www.docker.com/). It is designed to provide users with clean and orderly appearance of interface components, reducing the chances of faulty operations and improving user experience. It utilizes high-performance computing resources to guarantee efficient, sustainable, and reliable services throughout a high volume of tasks. All datasets, models and results are stored with unique user identification generated by G2PDeep, keeping the information private and retrievable for users even without logging. The architecture of G2PDeep consists of four modules shown in Figure 2 and their details are summarized below.

Web interface module. This module is designed to be user-friendly by using enterprise-level user interface (UI) libraries, such as Ant Design and React UI (https://ant.design/ and https://reactjs.org/). The interface is responsive, making its appearance virtually the same regardless of the screen size on a computer and tablet. The Highcharts (https://www.highcharts.com/), an interactive JavaScript multi-platform charting library, is used for data visualization on cross-platform web browsers including Google Chrome, Firefox, Microsoft Edge and Safari. The interactive charts not only facilitate user access to the most interesting portions of experimental results but also provide a comprehensive view to explore the results from all aspects.

Middleware module. This module is used to bridge the gap between the web interface and the database. The Django framework (https://www.djangoproject.com/), a Python-based server-side web framework, is used to provide robust and powerful services. The module utilizes restful API, which uses HTTP requests to access and retrieve data and model information. It uses Python data analysis libraries, such as Pandas, NumPy and SciPy (https://pandas.pydata.org/, https://numpy.org/ and https://www.scipy.org/), to validate uploaded files and create training, validation and test datasets. The module saves the MD5 message-digest algorithm for uploaded files and ignores those files with the same MD5 code when uploading data to the database. It is equipped with TensorFlow (https://www.tensorflow.org/) and Keras (https://keras.io/) open-source machine learning platforms, for construction, training and inference of deep-learning neural networks.

Core module. This module is developed primarily based on Celery (https://github.com/celery/celery/), a Python-based task scheduler library. The module mainly contains task queues, worker nodes, task schedulers and message brokers. The task queues are used as a mechanism to distribute the work across threads and the number of threads is detected automatically according to CPU cores with high-computing resources. The tasks of training and inference of deep-learning networks are executed concurrently on multiple workers using multiprocessing. The task scheduler communicates via messages with message brokers to mediate among workers, allowing high horizontal calling.

Database module. MySQL (https://www.mysql.com/) and Redis and (https://redis.io/) databases are used in this module. Using the advantage of a relational database, MySQL integrates various datasets, metadata of datasets, project information, performance of model and model hyperparameters, including learning rate, structure of model, epochs, etc. All private datasets and models are only visible and accessible to datasets uploaders, in order to prevent data from leaking to unauthorized parties. Redis is a NoSQL database and extremely fast in reading and writing operations because of data stored in primary memory. The module utilizes Redis to store the task information and details of the scheduler, bring the reliability of data storage and transactions during multiple tasks processing.

Methods to generate inputs

The input files for the server are genotype and quantitative phenotype files in comma-separated values (CSV) format. To generate the required format, users can utilize PLINK2 and VCFTools, for zygosity and SNP data respectively, to convert a Variant Call Format (VCF) (15) file into a tab-delimited text file. Users can use Pandas to filter out the unnecessary information from the tab-delimited text file and save it to a CSV file.

RESULTS

Evaluation and metrics

To demonstrate the performance of quantitative phenotype prediction of G2PDeep, we compared G2PDeep with four well-established statistical models such as DeepGS, rrBLUP, Bayesian ridge regression (BRR) and Bayesian with LASSO.

For SoyNAM dataset, it contains 5132 recombinant inbred lines (RILs) and 4236 SNPs. From 2013 and 2012 at Illinois Location, we selected five traits including grain yield, height, moisture, oil, and protein of the soybean dataset. We generated five datasets with the same genotype and five different quantitative phenotypes. The Bandillo's dataset consists of 52,041 SNPs scored on 12 000 *G. max* accessions using the Illumina Infinium SoySNP50K BeadChip. Currently available oil and protein content data from the USDA GRIN (https://data.nal.usda.gov/dataset/germplasm-resources-information-network-grin) was used as phenotype. Likewise, we generated two datasets with the same genotype and two different quantitative phenotypes. We independently trained and evaluated our model

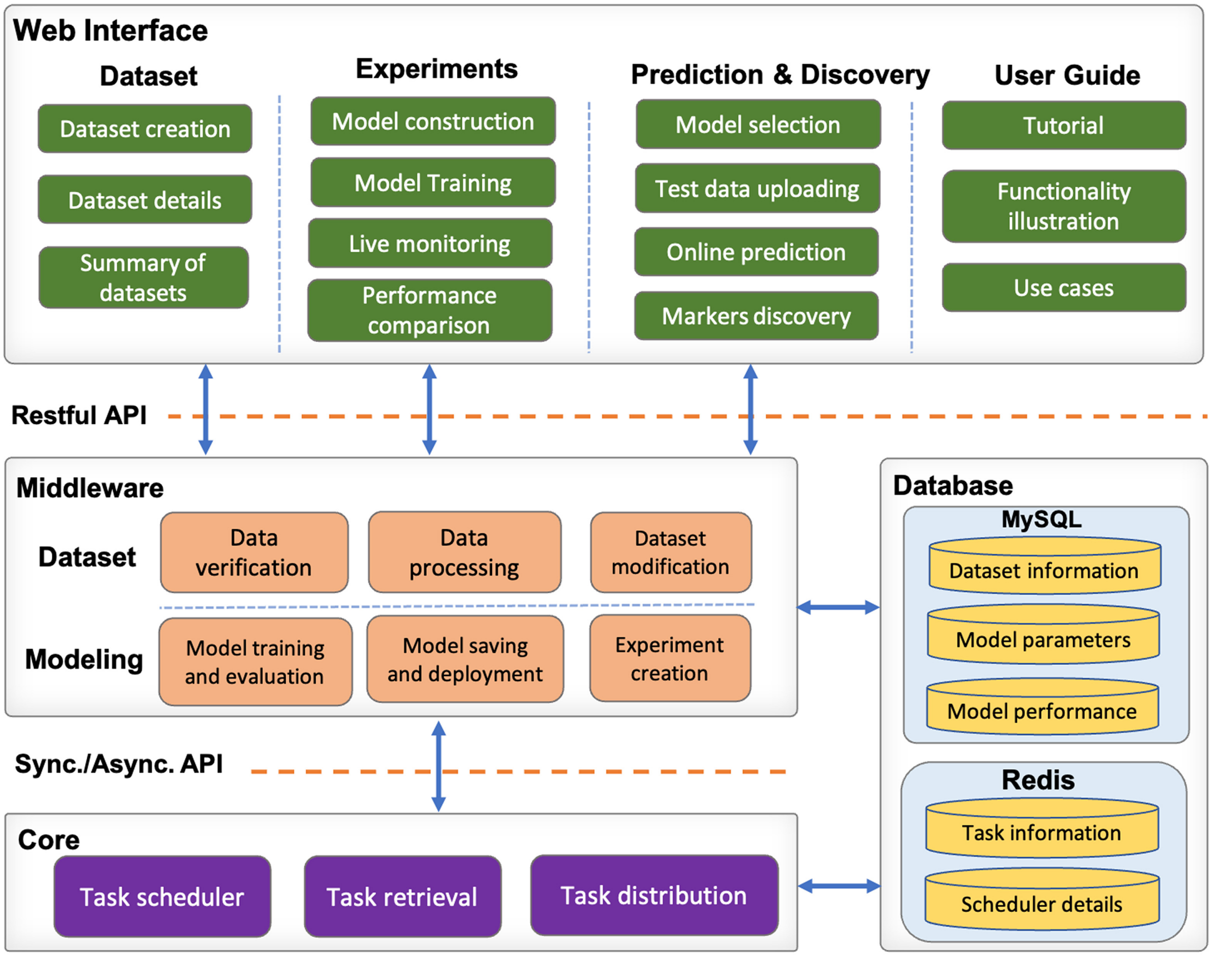


Figure 2. Illustration of G2PDeep architecture. The architecture consists of four modules and these modules communicate with each other via appropriate APIs.

Table 1. Pearson correlation coefficient of models on five datasets from the SoyNAM dataset. The dualCNN is the model currently used in G2PDeep. DeepGS is a model combining CNN and a fully connected neural network. rrBLUP is ridge regression with a relationship matrix and Gaussian kernel. BRR is Bayesian ridge regression. Bayesian LASSO is Bayesian regression with an L2 penalty.

	Pearson correlation coefficient				
Model	Yield	Protein	Oil	Moisture	Height
dualCNN	**0.448**	**0.624**	**0.654**	**0.453**	**0.611**
DeepGS	0.391	0.506	0.531	0.310	0.452
rrBLUP	0.412	0.392	0.39	0.413	0.458
BRR	0.422	0.392	0.39	0.413	0.458
Bayesian LASSO	0.419	0.394	0.388	0.416	0.458

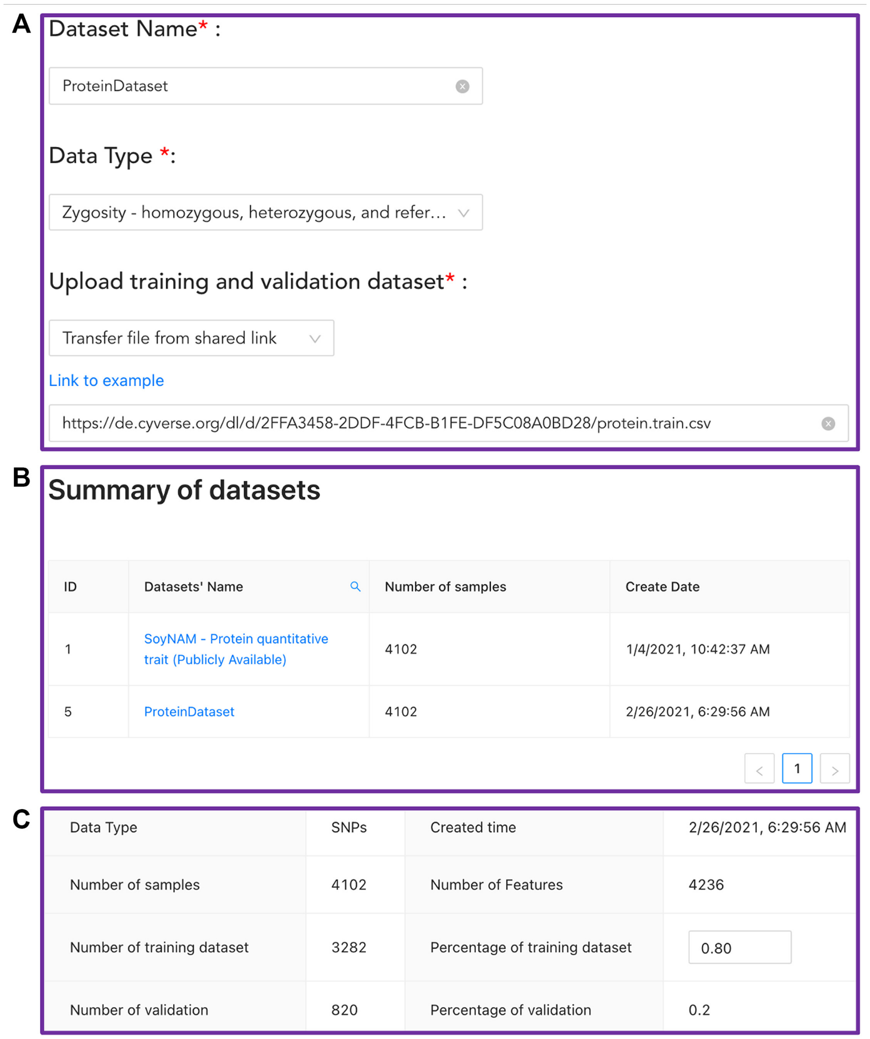
Table 2. Pearson correlation coefficient of models on two datasets from the Bandillo's dataset. The dualCNN is the model currently used in G2PDeep. The DeepGS is a model combining CNN and a fully connected neural network. rrBLUP is ridge regression with a relationship matrix and Gaussian kernel. BRR is Bayesian ridge regression. Bayesian LASSO is Bayesian regression with the L2 penalty.

	Pearson correlation coefficient	
Model	Protein	Oil
dualCNN	**0.467**	**0.674**
DeepGS	0.453	0.543
rrBLUP	0.434	0.533
BRR	0.443	0.521
Bayesian LASSO	0.412	0.534

using five datasets from SoyNAM and two datasets from the Bandillo's dataset.

None of the datasets was processed by any imputation. Each of the datasets was split randomly into training, validation and test datasets with the ratio of 3:1:1. The quantitative phenotype is normalized by a *z*-score. The performance of test data from SoyNAM and Bandillo's datasets was evaluated independently using the Pearson correlation coefficient, as shown in Tables 1 and 2, respectively.

To measure the performance of marker significance estimation, we compared the saliency map with a standard GWAS method using the 'gwas2' function of the 'NAM' R package (6) with the SoyNAM and Bandillo's datasets. The package estimates marker significance by negative log of the *P*-value. The marker significance estimated using saliency map and GWAS for SoyNAM and Bandillo's datasets is shown in Supplementary Figures S1 and S2, respectively. Furthermore, for each of the datasets, two sets of 100 mark-



Dataset Name* :

ProteinDataset

Data Type *:

Zygosity - homozygous, heterozygous, and refer...

Upload training and validation dataset* :

Transfer file from shared link

Link to example

https://de.cyverse.org/dl/d/2FFA3458-2DDF-4FCB-B1FE-DF5C08A0BD28/protein.train.csv

Summary of datasets

ID	Datasets' Name	Number of samples	Create Date
1	SoyNAM - Protein quantitative trait (Publicly Available)	4102	1/4/2021, 10:42:37 AM
5	ProteinDataset	4102	2/26/2021, 6:29:56 AM

< 1 >

Data Type	SNPs	Created time	2/26/2021, 6:29:56 AM
Number of samples	4102	Number of Features	4236
Number of training dataset	3282	Percentage of training dataset	0.80
Number of validation	820	Percentage of validation	0.2

Figure 3. Dataset creation and retrieval in G2PDeep. (**A**) An example of an uploading file by a shared link to data. Both dataset name and link are required. (**B**) The uploaded dataset and publicly available dataset are shown with metadata. (**C**) Details of the dataset including the number of features and number of SNPs in the training and validation datasets.

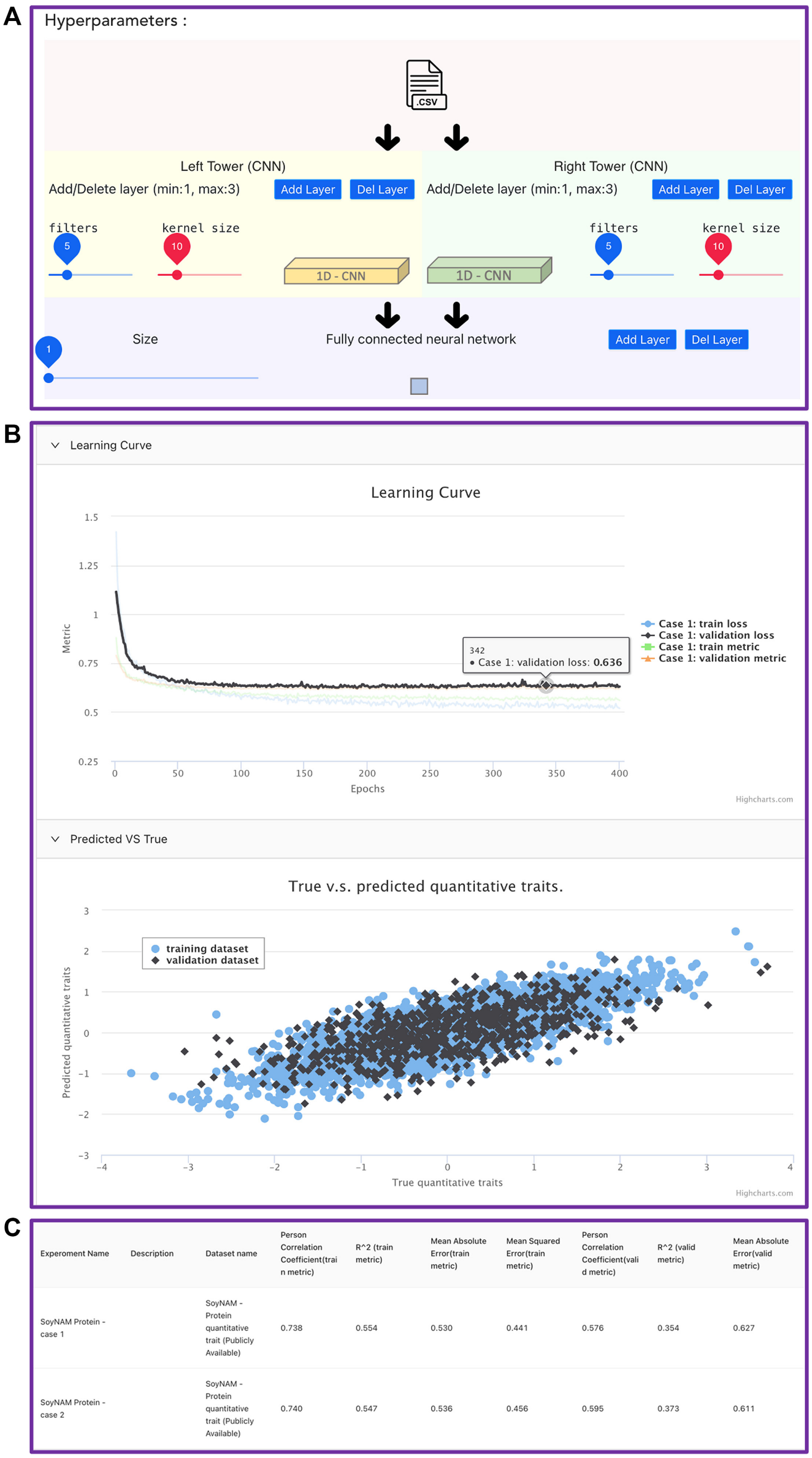
ers with the highest significance from saliency and GWAS are used to show the logical relations. The relationships are shown in Supplementary Figures S3 and S4, for SoyNAM and Bandillo's datasets respectively, which shows that markers with high significance in the saliency map also have a high significance in the GWAS.

Datasets in G2PDeep

The server allows users to upload and create datasets that are used to train and evaluate the model (see Figure 3A). It provides two options for input data, uploading a file directly or transferring data from a link. For a small-size dataset, users can upload a comma-separated values (CSV) file up to 50MB. For a large-size dataset, users can upload data by entering a shared link to the file to reduce data transfer time. The shared link can be generated using CyVerse Data Store, Google Drive, Dropbox and Microsoft OneDrive, and its file is parsed and downloaded directly via the server. The maximum dataset size allowed is 5GB. The server also provides the instruction and an example of data format aligned with the upload field. The uploaded dataset is validated according to data type and format. For an invalid dataset or file format, the server has functionalities to stop dataset creation and show corresponding error message. A progress bar is shown, allowing users to monitor the status of the dataset creation. The server integrates the publicly available SoyNAM and Bandillo's datasets and users are able to use them to train a model directly. The summary of the datasets is listed with dataset names and number of samples (see Figure 3B). Users can retrieve details of the datasets including number of features, and sizes of training and validation dataset (see Figure 3C). Sizes of training and validation datasets can be changed via the options on the website as well. The server takes ∼5 minutes on average to create a dataset with 1GB size file.

Deep-learning projects in G2PDeep

Conducting customizable deep-learning projects on a webserver is one of the key features in G2PDeep. In creating a project page (see Figure 4A), G2PDeep requires users to en-



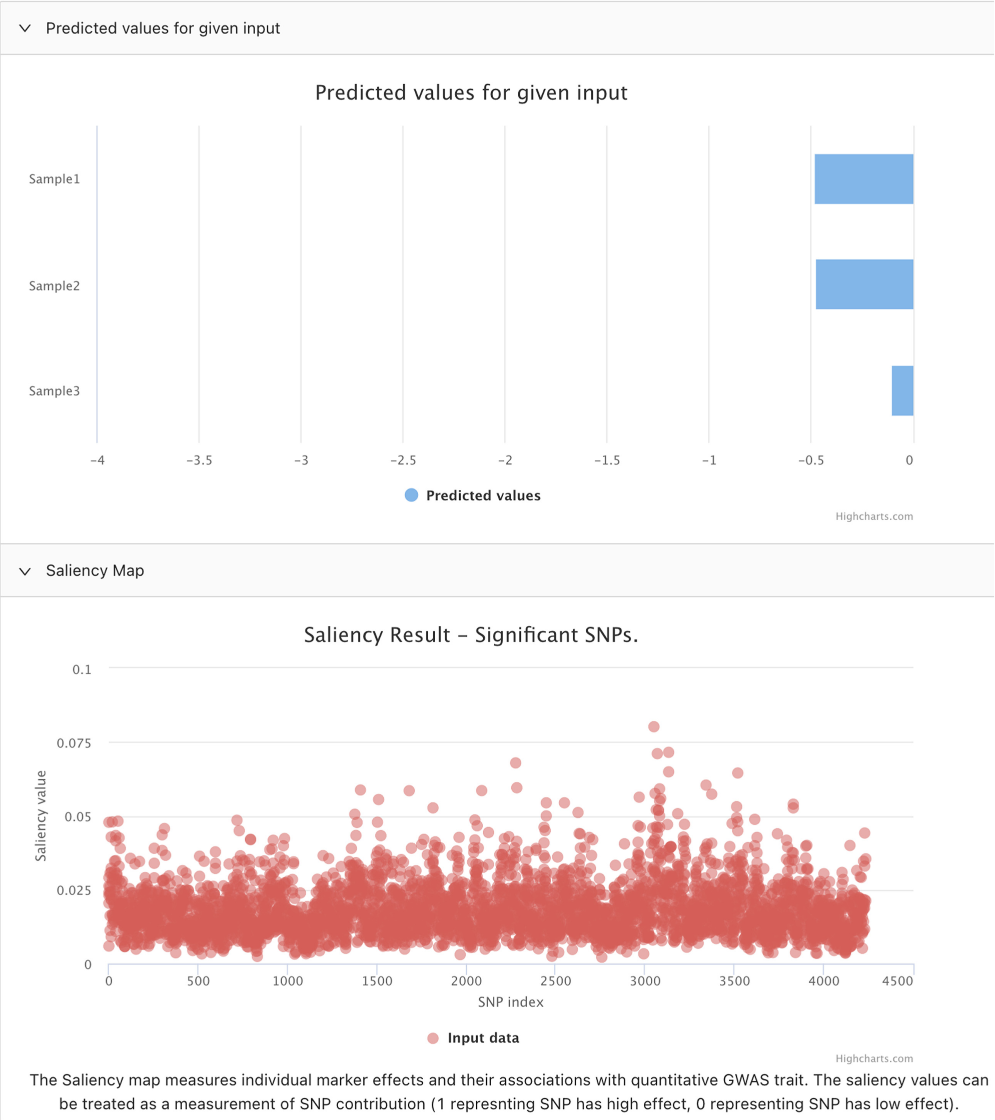
Left Tower (CNN): filters 5, kernel size 10

Right Tower (CNN): filters 5, kernel size 10

Fully connected neural network: Size 1

Experoment Name	Description	Dataset name	Person Correlation Coefficient(train metric)	R^2 (train metric)	Mean Absolute Error(train metric)	Mean Squared Error(train metric)	Person Correlation Coefficient(valid metric)	R^2 (valid metric)	Mean Absolute Error(valid metric)
SoyNAM Protein - case 1		SoyNAM - Protein quantitative trait (Publicly Available)	0.738	0.554	0.530	0.441	0.576	0.354	0.627
SoyNAM Protein - case 2		SoyNAM - Protein quantitative trait (Publicly Available)	0.740	0.547	0.536	0.456	0.595	0.373	0.611

Figure 4. Project section in G2PDeep. (**A**) Interactive chart to configure the deep-learning model. (**B**) Learning curve showing losses and metrics for each epoch. The scatter plot of predicted and true quantitative phenotypes for training and validation dataset. (**C**) An example of comparison results between two models. The Pearson correlation coefficient, *R*-squared, mean absolute error and mean squared error are shown in the form of a table.



The Saliency map measures individual marker effects and their associations with quantitative GWAS trait. The saliency values can be treated as a measurement of SNP contribution (1 represnting SNP has high effect, 0 representing SNP has low effect).

Figure 5. Results of predicted quantitative phenotypes and marker significance. The saliency map shows that the marker highly associated with phenotype is located in around 3000 SNP index.

ter the project name and appropriate description as a unique identification for future retrieval. Users can choose from the publicly available dataset or the private dataset created by themselves in the Datasets section. Users are also able to define the project settings by modifying parameters, such as learning rate, epoch and batch size. The architecture of the model is configurable in terms of hyperparameters to perform a reachability prediction. The number of CNN layers and fully connected neural network (DNN) can be changed independently using 'add' and 'delete' buttons. The size of filters and kernel size for each layer of CNN can also be changed separately using the sidebars. The number of neurons in DNN is configurable as well. The range of number of layers is restricted to 1 to 3. The early stopping method is applied by default to stop training when a monitored metric has stopped improving. This functionality significantly reduces the time required for training. Once the project is submitted, the project is placed in a task queue waiting for the computing resource allocation. The project settings and configuration of models are saved in the database. Using the CPU resource, the server takes about 50 minutes on average to train a customized model with 4000 training samples in 1000 epochs.

All these user-specific projects are retrievable on the summary page. The projects and their status are shown in the table. The status of the project is shown by a category such as pending, running, success and failure along with a progress bar showing the progress of model training. The details regarding dataset loading, parameters loading and model training can be seen by hovering the mouse over the status of the project. The estimated time remaining to complete the training process is also shown. All of the statuses of projects are updated automatically every ten seconds. The project name is linked to the corresponding details section (see Figure 4B), which provides a flow chart of the model, along with a number of parameters in each layer. The training parameters and hyperparameters are also listed in the table. In addition, a learning curve for training and validation datasets is shown during the training process, graphically depicting the strength of overfitting and underfitting. The scatterplot of the predicted and true quantitative phenotypes is displayed once the training process is done, intuitively showing their relationship. Users can observe the specific values using their mouse to hover over these charts.

G2PDeep allows users to compare the performance with up to four models. In the comparison section (see Figure 4C), the project details are listed as a table with loss and metrics including Pearson correlation coefficient, R-squared, mean absolute error, and mean squared error for both training and validation datasets. These losses and metrics are also displayed in a chart, providing users with exact values for each specific epoch.

Phenotype prediction and marker discovery

With the selection of a well-trained model and the correctly formatted input data, users can predict and visualize the results instantly. G2PDeep takes less than 30 seconds on average to predict quantitative phenotype and marker significance for 1000 samples of genotypes. For the input data, users can copy the genotype data from an excel file and paste it into the text field. Users can also upload a CSV file directly to the server. As usual, there is an example along with the Input Data field to illustrate the data format or simply load the publicly available SoyNAM dataset. The input data is validated according to the required data type and format. G2PDeep shows an error message for invalid input data. The maximum number of allowed samples is 10 000. G2PDeep also provides a progress bar for monitoring the status of prediction easily. After submission, G2PDeep provides a bar chart of predicted values and a saliency map (see Figure 5). In the saliency map, the particular value of a marker can be observed by using their mouse to hover over the data point. The predicted quantitative phenotypes and markers ranked by saliency values can be downloaded via the link in the chart.

User guide

We have developed a user guide with instructions about the functions and usage of G2PDeep website. The entire user guide is divided into four sections according to the main functions of G2PDeep – Introduction, Datasets, Projects, Prediction and Discovery. Both new and experienced users can find detailed instructions on how to utilize these functions. When users navigate the user guide on G2PDeep website, they can easily find demos for every single function. On each page, a link to a tutorial video is provided to show users how to follow different steps to run predictions using the server.

CONCLUSIONS AND FUTURE WORK

G2PDeep, to our knowledge, is the first web server for quantitative phenotype prediction and genomics marker discovery. It features a web-based framework with an interactive interface, enabling deep-learning models to be created, trained and inferenced using datasets uploaded by users. With an efficient and powerful middleware and core backend, the server can provide real-time training, model evaluation, live monitoring and inference for large-scale genotype data. G2PDeep provides users with interactive charts to show significant markers that are highly associated with a specific quantitative phenotype. Compared with other related works, G2PDeep has great performance in accuracy and scale. Considering the great potential of genomic selection in machine learning, G2PDeep would be a useful server in marker discovery associated with various phenotypes for plants and animals. The deep-learning model used in G2PDeep is developed as a publicly available stand-alone tool, enabling users to run G2PDeep on their local machine.

In the future, we plan to extend the server by applying automated machine learning (AutoML) to automatically adjust hyperparameters, eliminating the need for skilled data scientists to build deep-learning models. We will expand the server to support Variant Call Format (VCF) as input for SNP data. Our future efforts will also include regular updates to incorporate publicly available genotype datasets. We will also deploy G2PDeep on a server with both CPU and GPU resources to facilitate model training and inference. Currently, we are working on combining significant markers, that are highly associated with phenotype, with biological meaningful annotation such as Gene Ontology (GO) (16), Pfam protein families and domains (17) and KEGG Pathway (18) annotation. This will become available as a feature in the upcoming version.

DATA AVAILABILITY

G2PDeep is available as a web server at http://g2pdeep.org. The Python-based deep-learning model to conduct quantitative phenotype and marker discovery prediction on a local machine is available in the GitHub repository (https://github.com/shuaizengMU/G2PDeep_model).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

Missouri soybean farmers' checkoff dollars provided by the United Soybean Board (USB) 'Applied Genomics to Improve Soybean Seed Protein' Project [1920-152-0131-C to T.J.]; NSF Plant Genome Research Program Award [IOS-1444448 and IOS-1546873 to T.J.]; National Institutes of Health [R35-GM126985 to D.X.]. Funding for open access charge: Missouri soybean farmers' checkoff dollars provided by the United Soybean Board (USB) "Applied Genomics to Improve Soybean Seed Protein" Project.

Conflict of interest statement. None declared.

REFERENCES

1. Meuwissen,T.H., Hayes,B.J. and Goddard,M.E. (2001) Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, **157**, 1819–1829.
2. Endelman,J.B. (2011) Ridge regression and other kernels for genomic selection with R package rrBLUP. *Plant Genome*, **4**, 250–255.

3. Ma,W., Qiu,Z., Song,J., Li,J., Cheng,Q., Zhai,J. and Ma,C. (2018) A deep convolutional neural network approach for predicting phenotypes from genotypes. *Planta*, **248**, 1307–1318.
4. Schaeffer,L. (2006) Strategy for applying genome-wide selection in dairy cattle. *J. Anim. Breed. Genet.*, **123**, 218–223.
5. Spindel,J., Begum,H., Akdemir,D., Virk,P., Collard,B., Redona,E., Atlin,G., Jannink,J.-L. and McCouch,S.R. (2015) Genomic selection and association mapping in rice (Oryza sativa): effect of trait genetic architecture, training population composition, marker number and statistical model on accuracy of rice genomic selection in elite, tropical rice breeding lines. *PLoS Genet.*, **11**, e1004982.
6. Xavier,A., Jarquin,D., Howard,R., Ramasubramanian,V., Specht,J.E., Graef,G.L., Beavis,W.D., Diers,B.W., Song,Q. and Cregan,P.B. (2018) Genome-wide analysis of grain yield stability and environmental interactions in a multiparental soybean population. *G3: Genes Genomes Genet.*, **8**, 519–529.
7. Verbyla,K.L., Hayes,B.J., Bowman,P.J. and Goddard,M.E. (2009) Accuracy of genomic selection using stochastic search variable selection in Australian Holstein Friesian dairy cattle. *Genet. Res.*, **91**, 307–311.
8. Liu,Y., Wang,D., He,F., Wang,J., Joshi,T. and Xu,D. (2019) Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. *Front. Genet.*, **10**, 1091.
9. Szegedy,C., Ioffe,S., Vanhoucke,V. and Alemi,A. (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. *Proc. AAAI Conf. Artif. Intell.*, **31**, 4278–4284.
10. Simonyan,K., Vedaldi,A. and Zisserman,A. (2013) Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv doi: https://arxiv.org/abs/1312.6034, 19 April 2014, preprint: not peer reviewed.
11. Song,Q., Yan,L., Quigley,C., Jordan,B.D., Fickus,E., Schroeder,S., Song,B.H., Charles An,Y.Q., Hyten,D. and Nelson,R. (2017) Genetic characterization of the soybean nested association mapping population. *Plant Genome*, **10**, 1–14.
12. Bandillo,N., Jarquin,D., Song,Q., Nelson,R., Cregan,P., Specht,J. and Lorenz,A. (2015) A population structure and genome-wide association analysis on the USDA soybean germplasm collection. *Plant Genome*, **8**, https://doi.org/10.3835/plantgenome2016.10.0109.
13. Merchant,N., Lyons,E., Goff,S., Vaughn,M., Ware,D., Micklos,D. and Antin,P. (2016) The iPlant collaborative: cyberinfrastructure for enabling data to discovery for the life sciences. *PLoS Biol.*, **14**, e1002342.
14. Goff,S.A., Vaughn,M., McKay,S., Lyons,E., Stapleton,A.E., Gessler,D., Matasci,N., Wang,L., Hanlon,M. and Lenards,A. (2011) The iPlant collaborative: cyberinfrastructure for plant biology. *Front. Plant Sci.*, **2**, 34.
15. Danecek,P., Auton,A., Abecasis,G., Albers,C.A., Banks,E., DePristo,M.A., Handsaker,R.E., Lunter,G., Marth,G.T., Sherry,S.T. *et al.* (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.
16. Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S. and Eppig,J.T. (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.
17. Mistry,J., Chuguransky,S., Williams,L., Qureshi,M., Salazar,G.A., Sonnhammer,E.L.L., Tosatto,S.C.E., Paladin,L., Raj,S., Richardson,L.J. *et al.* (2020) Pfam: The protein families database in 2021. *Nucleic Acids Res.*, **49**, D412–D419.
18. Kanehisa,M., Furumichi,M., Sato,Y., Ishiguro-Watanabe,M. and Tanabe,M. (2020) KEGG: integrating viruses and cellular organisms. *Nucleic Acids Res.*, **49**, D545–D551.