BMC Bioinformatics

SOFTWARE

Open Access

Accurate plant pathogen effector protein classification *ab initio* with deepredeff: an ensemble of convolutional neural networks

Ruth Kristianingsih and Dan MacLean*

*Correspondence:
dan.maclean@tsl.ac.uk
The Sainsbury Laboratory, University of East Anglia, Norwich, UK

Abstract

Background: Plant pathogens cause billions of dollars of crop loss every year and are a major threat to global food security. Effector proteins are the tools such pathogens use to infect the cell, predicting effectors de novo from sequence is difficult because of the heterogeneity of the sequences. We hypothesised that deep learning classifiers based on Convolutional Neural Networks would be able to identify effectors and deliver new insights.

Results: We created a training set of manually curated effector sequences from PHI-Base and used these to train a range of model architectures for classifying bacteria, fungal and oomycete sequences. The best performing classifiers had accuracies from 93 to 84%. The models were tested against popular effector detection software on our own test data and data provided with those models. We observed better performance from our models. Specifically our models showed greater accuracy and lower tendencies to call false positives on a secreted protein negative test set and a greater generalisability. We used GRAD-CAM activation map analysis to identify the sequences that activated our CNN-LSTM models and found short but distinct N-terminal regions in each taxon that was indicative of effector sequences. No motifs could be observed in these regions but an analysis of amino acid types indicated differing patterns of enrichment and depletion that varied between taxa.

Conclusions: Small training sets can be used effectively to train highly accurate and sensitive deep learning models without need for the operator to know anything other than sequence and without arbitrary decisions made about what sequence features or physico-chemical properties are important. Biological insight on subsequences important for classification can be achieved by examining the activations in the model

Keywords: AI, Deep learning, Effector protein

Background

Phytopathogens are a major threat to global crop production. The fungal phytopathogen *Magnoporthe oryzae* that causes cereal blast is responsible for around 30% of rice production loss and has now emerged as a pandemic problem on wheat [1] The oomycete *Phytophthora infestans* causes losses of around 6 billion USD to potato production,

© The Author(s), 2021. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

annually [2]. The bacterium *Ralstonia solanacearum* has a wide host range and can cause loses of over 30% in potato, banana and groundnut [3]. The incidences of crop disease are increasing, global climate change and agricultural practice are expanding the geographical range of pathogens and upping the stakes in the evolutionary arms race. Effector proteins are the shock troops of infection, manipulating the host at the infection interface to the pathogens advantage. Identifying and characterising a pathogen's effector content is a critical first step in understanding diseases and developing resistance, but effectors are notoriously difficult to characterise from sequence data. In most phyla they have only a few easily determined sequence characteristics (some in fungi are cysteine rich or have a MAX motif, some in oomycetes have the RXLR motif or WY fold) but in many cases no sequence identifiers are known [4]. Characterising effectors requires painstaking molecular experimental work and genome-scale approaches have relied on complex computational pipelines with in-built a priori assumptions about what might constitute an effector sequence in the absence of sequence features known to group them [5]. To understand infection processes, to provide genome-level understanding of the functions of this important class of genes and to develop future disease resisting crop varieties there is a need to identify effectors computationally from genome and protein sequence data.

Machine learning (ML) algorithms are a general group of techniques most often used for classification of data into groups. Supervised ML require a set of training examples and associated data with which to learn. Defining the best data to use and collect, called feature selection is an important and difficult prerequisite. ML approaches have been applied with success to biological sequence analysis, particularly in transcription factor binding site prediction, for the classification of eukaryote and bacterial nuclear proteins [6] and in the plant pathogen domain work by Sperschneider et al. [7] developed two ensemble-based machine learning models that could identify effectors and predict localisation with $>$ 70% accuracy [8, 9].

Deep learning models are distinct from other machine learning processes in that preselection of important features is far less critical and the models can learn these features unsupervised from training data [10]. This property removes the need to know which properties of a data set must be examined before data collection begins. The Deep learning models can therefore classify on properties not necessarily known to the operator and could be used to uncover cryptic patterns in data. Convolutional neural networks (CNNs) are a type of neural network that have found wide application in numerous machine vision problems, including image object classification and facial identification [11, 12], in time-series data analysis [13] and natural language processing [14]. In the biomedical domain they have been used in drug discovery [15] and gene network prediction [16]. In studies with bacterial type III secreted effectors Xue et al. developed an accurate CNN classifier for bacterial sequences [17]. CNNs encode information about the features used to classify that can be extracted and interpreted. In a sequence classification problems this means they have the potential to reveal novel sequence features that other bioinformatics approaches have not and could be of particular utility when analysing sets of effectors.

Deep learning approaches require positive and negative examples from which to learn. We used a list of sequences annotated as an effector or not. It is generally held

and expected that he larger and more accurate the list the more sensitivity a model can obtain. It is critical that training examples are experimentally verified effectors. Much of the effector annotation in public genomics databases is from computational predictions of genomics and is therefore of experimentally unverified hypothetical effectors. A good source of experimentally verified data is in the Molecular Plant Microbe Interactions (MPMI) literature and The widest ranging manual curation of MPMI papers is being performed as part of the PHI-Base [18] database strategy, PHI-base is an expertly curated database of genes proven experimentally to affect the outcome of pathogen host interactions and is therefore an excellent source of reliable effector sequences.

Here we use combinations of CNNs that do not rely on a priori feature selection to classify experimentally verified effectors and non-effectors in three taxa of plant pathogen: bacterial, fungal and oomycete. We show that these have very strong predictive power, can outperform existing effector prediction methods in accuracy and a better balance of sensitivity and specificity. We also analyse the activations of the models in response to effectors and non-effectors to gain insights into the sequence features that are allowing classification. We have produced an R package that will allow other scientists to easily classify their own sequences of interest, available at https://ruthkr.github.io/deepredeff.

Sequence data were collected from the PHI-Base database version 4.8 [18] by accessing a text dump of the data prepared on request by the PHI-Base team, the file can be accessed at https://github.com/PHI-base/data/blob/master/releases/phi-base_current.csv. The pipeline in Fig. 1 outlines the steps used. We filtered plant effector proteins and their taxonomic groups and collected sequences from UniProt Release 2019_05, using the code in https://github.com/TeamMacLean/ruth-effectors-prediction/blob/master/scripts/r-scripts/getting-data-new/binary-class/0001_first_step_getting_data.Rmd. We created a correspondingly sized data set of non-effectors with secretion signals originating in species matched to those from which the effectors were drawn. We downloaded sequences for randomly selected proteins matching these criteria from Ensembl databases [19]: specifically Ensembl Fungi, Protists and Bacteria manually using the BioMart tools [20]. Since the BioMart tool is not available on Ensembl Bacteria, we downloaded whole proteome protein sequnces from species matched to those from which the effector came using FTP. With these we used SignalP 3.0 [21] in order to filter the secreted sequences and selected accordingly. We used default paramaters from SignalP, except the type of organism group which is `euk` for both fungi and oomycete sequences, and `gram-` or `gram+` for bacteria sequences. Redundant sequences were filtered using BLASTp [22]. We achieved these steps using the code in https://github.com/TeamMacLean/ruth-effectors-prediction/blob/master/scripts/r-scripts/getting-secreted-data/0005_process_signalp_data.Rmd.

Encoding and subsetting sequences

The sequences collected were encoded using either one-hot encoding (CNN-LSTM based models) or integer based encoding (CNN-GRU-LSTM models). Sequences were post-padded with zeroes to bring the vectors to identical lengths to each other and the longest sequence in the taxon data set. The longest sequence for bacteria, fungi, and oomycete are 2574, 4034, and 934, respectively. Encoded sequences were split into

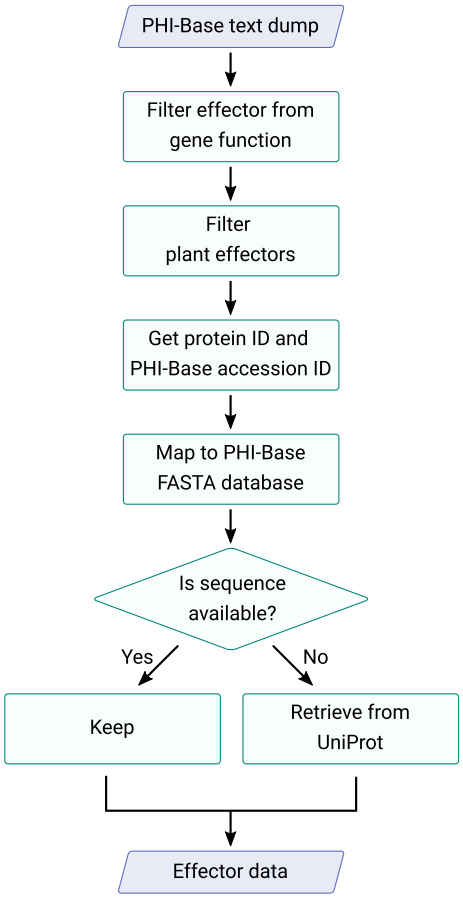

Fig. 1 Workflow diagram for collection of effector sequences from the PHI-Base database annotation and cross reference to UniProt

random taxon specific training, test and validation sets at a 60%, 20%, 20% split respectively as described in code at https://github.com/TeamMacLean/ruth-effectors-prediction/blob/master/scripts/r-scripts/getting-secreted-data/0008_split_and_encode.Rmd.

Model training

We trained four model types on each taxon specific sequence set: CNN-LSTM, CNN-GRU, LSTM-Embedding, GRU-Embedding. We trained each model using a basic random hyperparameter setting initialisation step followed by hyperparamter scans. All models were implemented in Python 3.6.9 [23] using the deep learning API Keras 2.2.4 [24] with the Tensorflow 1.12.0 [25] backend, using NVIDIA GK110GL Tesla K20c GPUs and AMD Opteron(TM) Processor 6272 CPUs with 128 GB RAM.

Hyperparameter scans

Hyperparameter scans were performed using random search, a hyperparameter optimization method where each hyperparameter setting is randomly sampled from a distribution of possible hyperparameter values [26]. Hyperparameters to vary were selected as the ones generally known and expected to have strongest effect. We used RandomSearchCV(), the implementation of random search in `scikit-learn 0.19.2` [27] together with

KerasClassifier() which is an implementation of the scikit-learn classifier API for keras. Code for this can be found in https://github.com/TeamMacLean/ruth-effectors-prediction/tree/master/scripts/python-scripts/hyperparameter-scan-scripts. All model training was performed as described in section Model Training.

Fine tuning

Fine tuning was performed manually using `keras 2.2.4` together with metrics module and KFold cross validator from `scikit-learn 0.19.2`. Scripts implementing the tuning can be found at https://github.com/TeamMacLean/ruth-effectors-prediction/tree/master/scripts/python-scripts/manual_tune_scripts. Five-fold cross validation was used in all instances.

Model classification correlation

We calculated correlations between the classifications from best performing models on the hold-out test data set using Pearson's correlation co-efficient on the 1/0 classification vectors.

Ensemble functions

We computed an aggregate classification using two different ensemble functions, weighted average and an overall majority option.

Weighted average is computed as

$$\bar{y} = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i}, \tag{1}$$

where w_i is the weight, y_i is the prediction value of the i^{th} model, and n is the total number of model. In our case, we use the accuracy of each model as the average.

Overall majority is computed as

$$\tilde{y} = argmax(x_1, x_2, \ldots, x_n), \tag{2}$$

where

$$y = \begin{cases} 1 & \text{for } \tilde{y} > 0.5 \\ 0 & \text{for } \tilde{y} \leq 0.5 \end{cases} \tag{3}$$

Metrics

We used the following calculations for different accuracy metrics in our evaluations, specifically: accuracy, sensitivity, specificity. *TP*, *TN*, *FP*, and *FN* refer to the number of true positives, true negatives, false positives and false negatives, respectively.

Accuracy (*Acc*) is the ratio between correctly classified non-effectors and effectors and all samples:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Sensitivity *Sn* is the ratio between correctly predicted as effectors and all effectors:

$$Sn = \frac{TP}{TP + FN} \tag{5}$$

Specificity (Sp) is the ratio between correctly predicted as non-effectors and all non-effectors:

$$Sp = \frac{TN}{TN + FP} \tag{6}$$

F1-score is the harmonic average of the precision and recall:

$$F_1 = \frac{2TP}{2TP + FN + FP} \tag{7}$$

Activation map analysis

To visualise the regions of the sequences that the models were using to discriminate between effector and non-effector we adapted a Grad-CAM (Gradient-weighted Class Activation Mapping) approach [28]. We extracted the convolutional layer `convd_1` and computed the gradient of the model output with respect to the output of `convd_1`. The feature map was weighted by every channel and mean result taken. All activation maps were summed, normalised and smoothed using Fourier analysis. We used `tensorflow 1.12.0` to compute the activation maps and discrete fourier transform (numpy.fft) from `numpy 1.17.3` to smooth the result. The code we used for computing and visualising these heatmaps can be found at https://github.com/TeamMacLean/ruth-effectors-prediction/tree/master/scripts/python-scripts/heatmaps .

Effector prediction software and training data

To test the performance of our models against commonly used tools we used DeepT3 version 1 [17] and EffectiveT3 version 1.0.1 [29] for bacterial sequences. Effector P Version 1.0 and 2.0 [8, 9] for fungal sequences and EffectR [30] for oomycete sequences. All the models publish the datasets used to train the models or some examples. We used positive training examples (effector sequence examples) in the comparisons we performed. EffectorP provides three different positive datasets (training, test and hold-out validation) for EffectorP 1.0 and 2.0 at http://effectorp.csiro.au/data.html. EffectiveT3 provides a training set at https://effectors.csb.univie.ac.at/sites/eff/files/others/TTSS_positive_training.faa. DeepT3 provides three sets, a non redundant Pseudomonas syringae effector dataset, a training dataset and a test data set at https://github.com/lje00006/DeepT3/tree/master/DeepT3/DeepT3-Keras/data. EffectR uses 6 RXLR oomycete sequences described in [2] as examples rather than as a training set, namely *PexRD36*, *PexRD1*, *ipi01/Avrblb1*, *Avr1*, *Avr4*, and *Avr3a* . All these sets were used in their respective tools with default settings.

Software implementation

To make our models useful for developers in their own analytic pipelines we have provided an R package that provides a useful interface to the models. The package

and installation instructions are available from GitHub https://ruthkr.github.io/deepredeff and CRAN https://cran.r-project.org/package=deepredeff

Results

Sequence collection

The performance of the trained classifiers is dependent on the quality of the input training data, so it was important that we collected as high a quality set of annotated effectors as possible. To this end we used PHI-Base [18] as our primary sequence origin. Sequences in PHI-Base are human curated from the literature and have therefore been noted in experimental studies. They do not derive from large scale annotations or contain hypothetical or predicted proteins. This attribute makes it ideal for our purposes as the effectors in PHI-Base are those that have been specifically reported as such in the published literature and are not of the class of sequences that are merely suspected of being effectors on the basis of carrying a secretion signal. To collect effector sequences we parsed a whole database text dump of version 4.8 https://github.com/PHI-base/data, all proteins marked as plant pathogen effectors were filtered and we used the IDs and UniProt IDs to collect the protein sequences from PHI-Base or UniProt if PHI-Base stored only the ID (see Fig. 1. The sequences and IDs retrieved can be seen in the data file in this manuscript's repository https://github.com/TeamMacLean/ruth-effectors-prediction/blob/master/data/getting-data-new/binary-class-data/effector_data.csv. Effector sequences were then divided into taxonomic groups as bacterial, fungal or oomycete derived accordingly. In total 190 bacterial effectors from 13 species were collected, 97 fungal effectors from 16 species were collected and 85 oomycete effectors from 6 species were collected (Table 1). The species and effector count in each group can be seen in Additional file 1: Tables S1-S3.

Sequences for non-effector, secreted proteins were collected using a similar pipeline. Randomly selected proteins from each species carrying secretion signals were extracted from Ensembl databases using the BioMart tool. For each species noted in Additional file 1: Tables S1-S3 we collected from either the same strain or species an identical number of non-effector, secreted proteins to that in the effector set. This gave us a balanced data set of effector proteins as positive learning examples and non-effector secreted proteins as negative learning examples. Figure 2 summarises the process of building the non-effector set, and the full set of sequences and IDs retrieved can be seen in the following data file https://github.com/TeamMacLean/ruth-effectors-prediction/tree/master/data/secreted_data.

Table 1 Count of effectors listed in publications curated by PHI-Base used in this study in three major plant pathogen groups

Pathogen group	Species	Effectors found
Bacteria	13	190
Fungi	16	97
Oomycete	6	85

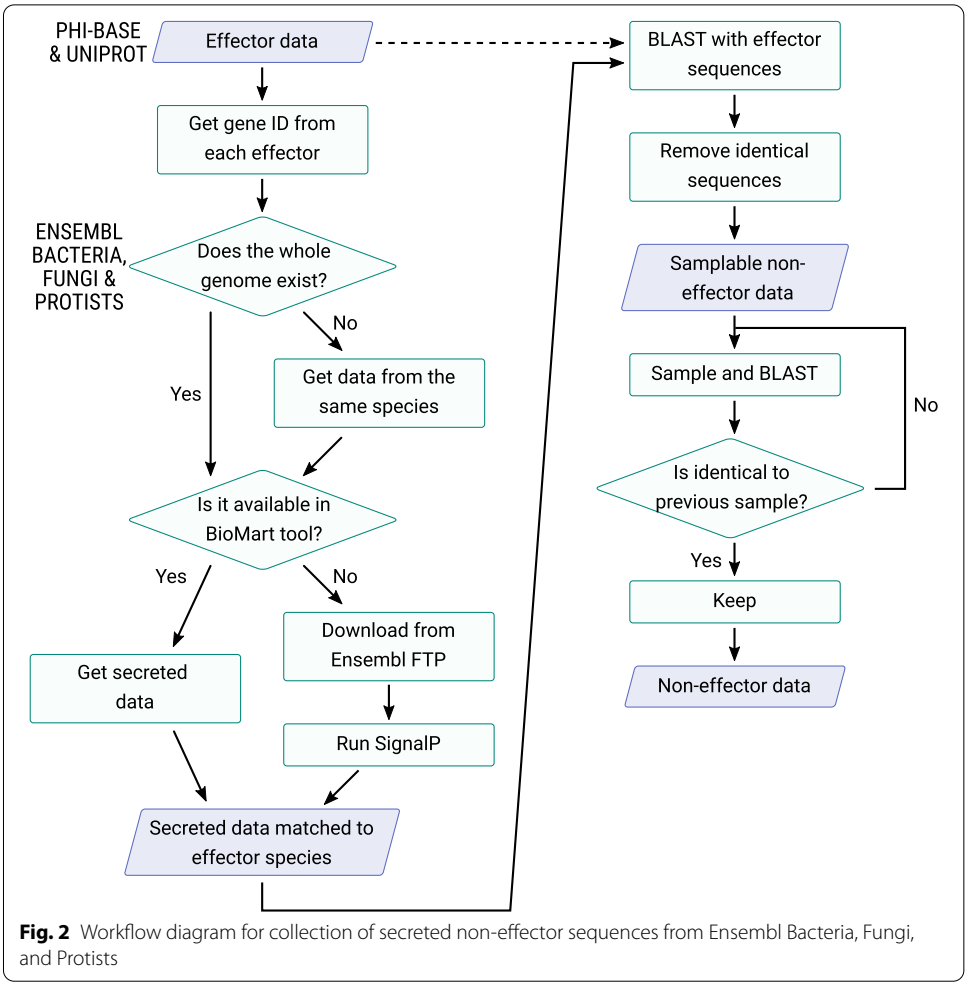

Fig. 2 Workflow diagram for collection of secreted non-effector sequences from Ensembl Bacteria, Fungi, and Protists

Model selection and training

In order to identify a useful classifier we took a randomised hyperparamter search over some likely base model architectures. We selected four base architectures on which to build models for learning. Two of these contained Convolutional Neural Network (CNN) layers followed by either a Long Short Term Memory Layer (LSTM) or a Gated Recurrent Unit (GRU), two contained an Embedding Layer followed by the LSTM or GRU. All models had fully-connected dense layers after this. See Fig. 3.

We defined a range of values for the hyperparameters that could be optimised in each architecture, 10 for CNN-LSTM, 12 for CNN-GRU, 9 for LSTM-Embedding and 9 for GRU-Embedding. To test all combinations of values in these ranges would take a prohibitive amount of processor time, so we selected 50 sets of values for each model in each taxon at random to start training, 3000 models in total. Model variants within the hyperparameter search were assessed by comparing accuracy values on the development validation fraction of the training data. Other hyperparameters were fixed and are listed in Additional file 1: Table S5. For each model type and taxon training data combination we selected the hyperparameter set giving highest accuracy on the validation fraction. From this we had twelve candidate models to develop further.

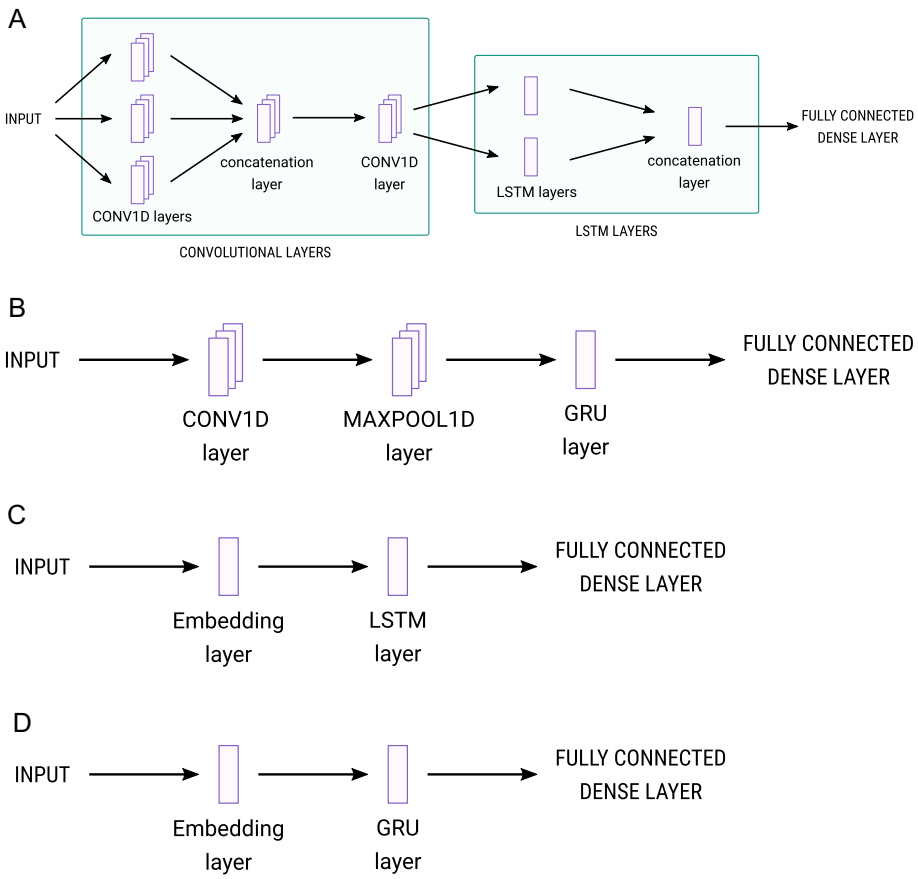

Fig. 3 Overview of model architectures tested in this analysis, **A** CNN-LSTM model; **B** CNN-GRU model; **C** LSTM-Embedding model; **D** GRU-Embedding model

We then manually ran and checked the accuracy and loss of the twelve models on the training and validation sets to investigate instances of overfitting and assess generality. Smaller models are less likely to overfit data, so we investigated the effect of regularization rate, filter count and kernel size on the progress and accuracy of the model as we reduced the size. Parameters varied in this phase are listed in S6. Final selected hyperparameter settings for models in each taxon can be seen in Additional file 1: Table S7. The values of accuracy and loss of each model produced are shown in Table 2. We found that by reducing the number of kernels on all models from 2 to 1 and the number of filters reduced from 32 to 16 we removed apparent overfitting and retained high accuracy, with training completing in 40 epochs.

Final training progressions for each model in each taxon can be seen in Fig. 4. We tested the finalised models on the hold-out test fraction of the data that had not been previously seen, for the four bacterial sequence trained models we had accuracies in the range 93.4 to 97.4, for the four fungal models we observed accuracy in the range 60.5 to 84.2 and for the four oomycete models we observed accuracy from 64.7 to 82.3, reported in Fig. 5. All the models we generated had high statistical power and can accurately and reliably classify effectors from other secreted proteins in that taxon.

Table 2 Accuracy and loss values from best performing parameters values for each model

Group	Data set	Model	Accuracy	Loss
Bacteria	Training	CNN-LSTM	0.917	0.334
Bacteria	Training	CNN-GRU	0.965	0.148
Bacteria	Training	LSTM-Emb	0.978	0.129
Bacteria	Training	GRU-Emb	0.947	0.202
Bacteria	Validation	CNN-LSTM	0.855	0.377
Bacteria	Validation	CNN-GRU	0.947	0.205
Bacteria	Validation	LSTM-Emb	0.934	0.232
Bacteria	Validation	GRU-Emb	0.895	0.385
Fungi	Training	CNN-LSTM	0.907	0.362
Fungi	Training	CNN-GRU	0.814	0.434
Fungi	Training	LSTM-Emb	0.771	0.505
Fungi	Training	GRU-Emb	0.788	0.484
Fungi	Validation	CNN-LSTM	0.526	0.923
Fungi	Validation	CNN-GRU	0.579	0.799
Fungi	Validation	LSTM-Emb	0.711	0.504
Fungi	Validation	GRU-Emb	0.842	0.468
Oomycete	Training	CNN-LSTM	0.657	0.626
Oomycete	Training	CNN-GRU	0.833	0.429
Oomycete	Training	LSTM-Emb	0.863	0.314
Oomycete	Training	GRU-Emb	0.706	1.487
Oomycete	Validation	CNN-LSTM	0.735	0.761
Oomycete	Validation	CNN-GRU	0.618	0.677
Oomycete	Validation	LSTM-Emb	0.588	0.910
Oomycete	Validation	GRU-Emb	0.500	1.486

The final twelve models were saved into HDF5 objects and stored in the repository at https://github.com/TeamMacLean/ruth-effectors-prediction/tree/master/data/final_model_hdf5.

Model characteristics

We examined the tendency of the models to call false positives or false negatives preferentially by creating confusion matrices of the classifications relative to the ground truth on the hold-out test data. The bacterial sequence trained models in general showed high accuracy and only one or two miscalls with no error bias except for the GRU-Embedding model which called five from 38 effectors as non effectors. The fungal sequence trained models were less accurate overall and showed a small amount more bias, again in the GRU-Embedding model, which was biased towards calling effectors as non-effectors and the CNN-LSTM model which was slightly biased in the opposite direction, calling non-effectors as effectors. The oomycete models were again quite balanced but the GRU-Embedding model showed a quite conservative tendancy calling 12 out of 17 effectors as non-effectors whilst getting all 17 non-effectors correct. Overall the models are accurate and show little to no bias toward false positive or false negatives, with the exception of the GRU-Embedding type. In oomycete sequences in particular and in this

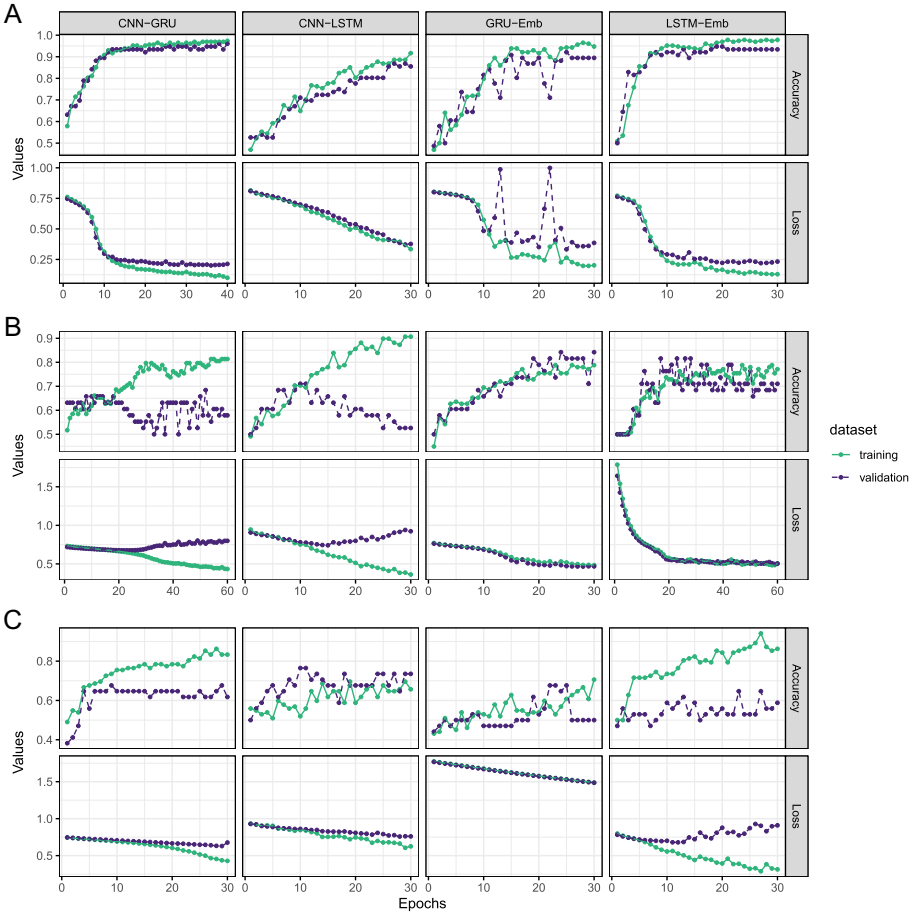

Fig. 4 Training trajectories showing Accuracy and Loss over 40 epochs for four model types on **A** Bacterial sequence training set; **B** Fungal sequence training set; **C** Oomycete sequence training set

class of model across the different sequence types showed itself to tend to call real effectors as not of that class.

Classification correlations between the different model architectures were high and positive in the bacterial sequence trained model's calls, in the range 0.8 to 0.88, see Fig. 6. CNN-GRU and LSTM-Embedding showed identical prediction sets. We observed similar levels of correlation in the CNN-LSTM, GRU-Embedding and CNN-GRU fungal sequence trained model, in the range 0.79 to 0.9 ; though there was a significantly lower range of correlations with the LSTM-Embedding which were in the range 0.36 to 0.51. The models trained on oomycete sequences all showed this lower range of correlations, in the range 0.30 to 0.65. The higher correlation across bacterial trained models is likely from a mixture of the larger training set size and a greater uniformity of the sequences themselves. For the fungal sequence trained models we can see that the LSTM-Embedding model does not perform as well as the others. The oomycete sequence trained models all show a lower range correlation reflecting the likely less uniform and smaller training set. It is clear that, particularly for the

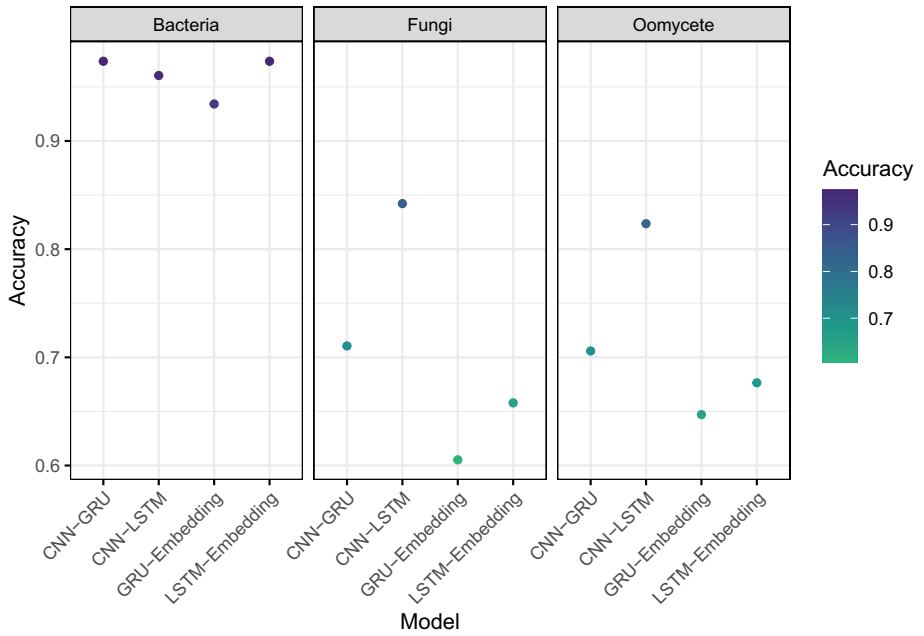

Fig. 5 Final performance of models in each taxon on the unseen, hold-out test data fraction of sequences

fungal and oomycete models, each architecture is capturing separate aspects of the sequences and classifying on those with slightly varying levels of success.

Ensemble models

We examined the usefulness of combining the predictions of the different model architectures using an ensemble function that takes the vectors of classifications of each model architecture as input. We performed the classification of the hold-out test data set using the ensembled models and the results can be seen in Fig. 7. With the models trained in bacterial sequences we observed an increase in classification accuracy over the best model, up to 0.99 for both ensemble functions. However, with the fungal and oomycete models we observed decreases relative to the best single model in both cases due to the higher accuracy of the CNN-LSTM model being diluted by the combined inacuracy of the other model architectures. Examining the overlaps in classifications between the CNN-LSTM and CNN-GRU/LSTM-Embedding respectively showed that the two lesser performing models were not simply predicting subsets of the CNN-LSTM model, in both cases the lesser models were able to identify three effectors correctly that were missed by the generally stronger models. This indicates that the weaker models may be classifying on some patterns missed by the CNN-LSTM model.

Comparison with other classification software

Given the accuracies of the above we selected the ensemble bacterial model and the CNN-LSTM fungal and oomycete models to evaluate the performance of our models against widely used effector identification software. We compared against predictions from the bacterial effector prediction programs DeepT3 [17] and EffectiveT3 classification module for plant-associated bacteria 1.0.1 [29], the fungal effector prediction

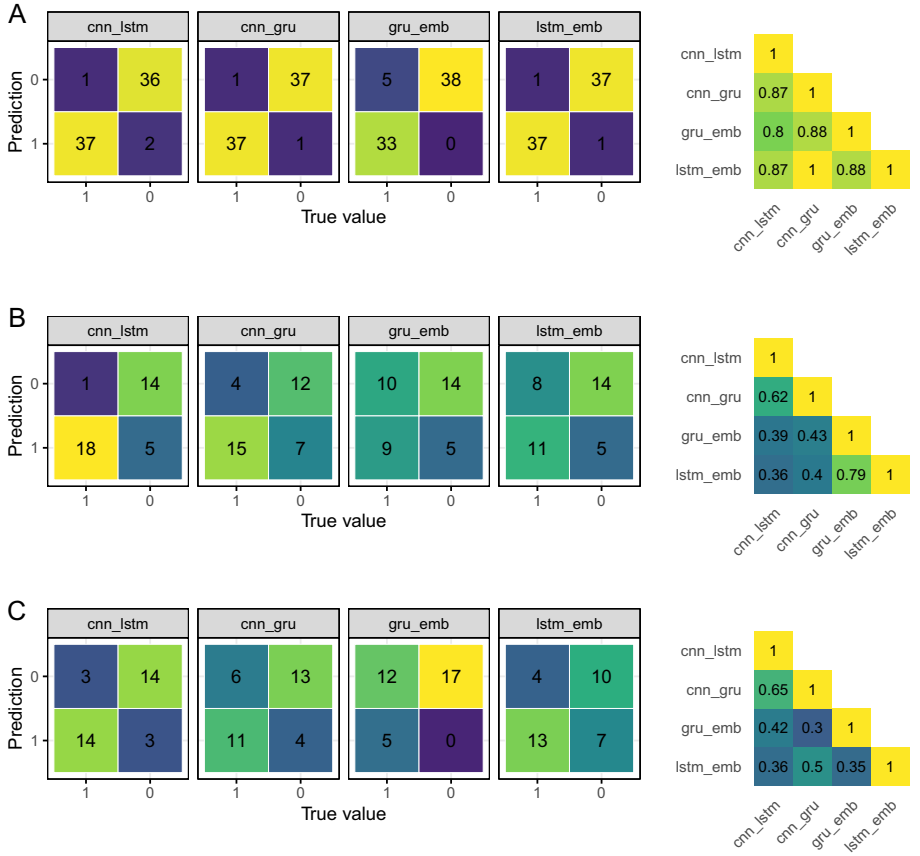

Fig. 6 Confusion matrices showing tendency of models to call false positive or false negative errors on the final test data; and Pearson correlations between classifications made by different models. **A** Models trained on bacterial sequences; **B** Models trained on fungal sequences; **C** Models trained on oomycete sequences

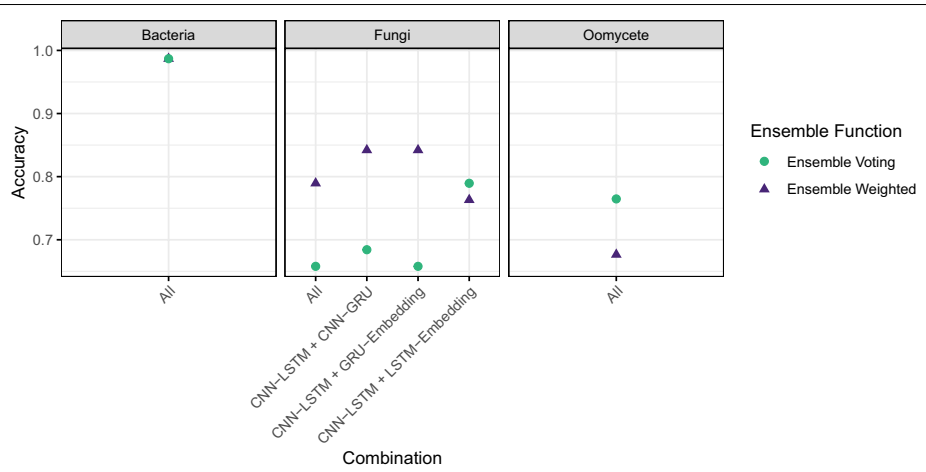

Fig. 7 Final performance of ensemble models in each taxon on the unseen, hold-out test data fraction of sequences

programs EffectorP 1.0 and 2.0 [8, 9] and the oomycete effector predictor EffectR [30]. Each comparison was carried out using the respective hold-out test sequence set for each taxon. For all taxa we observed greater Accuracy and F1 scores from our models than the established software, as shown in Fig. 8. This was particularly marked in the F1 score, which incorporates a measure of the incorrect calls. Absolute improvements were up to 15 % in bacterial sequences, 15 % in fungal sequences and 20% in the oomycete sequences. The confusion matrices in Fig. 9 show that accuracy and F1 score was compromised in all the established tools by the tendency of them all to misclassify true effectors as not effectors. All the established software classifiers we tested show lower sensitivity than the models we have developed here.

We also evaluated the deep learning models we have developed on the training and hold-out validation sequences used to train the previous methods. We calculated the proportion of the effectors in the training set that the tools could find on their respective training and validation sets, according to availability. The bacterial tools EffectiveT3 and DeepT3 showed lower proportion found than our Ensemble Weighted model, as in Fig 10A, consistent with the observation that our Ensemble Weighted model performed more strongly on the validation set that we generated. Interestingly, both versions of EffectorP found a greater proportion of the effectors in the EffectorP provided training sets than our CNN-LSTM model, but in the unseen validation data provided with EffectorP 2, all three models performed identically (Fig. 10B). The EffectorP 1 and 2 scores on validation data are well below the scores for the training data, a result that is usually interpreted as being evidence of an overfitted and less generalisable model. Our CNN-LSTM model for fungi showed

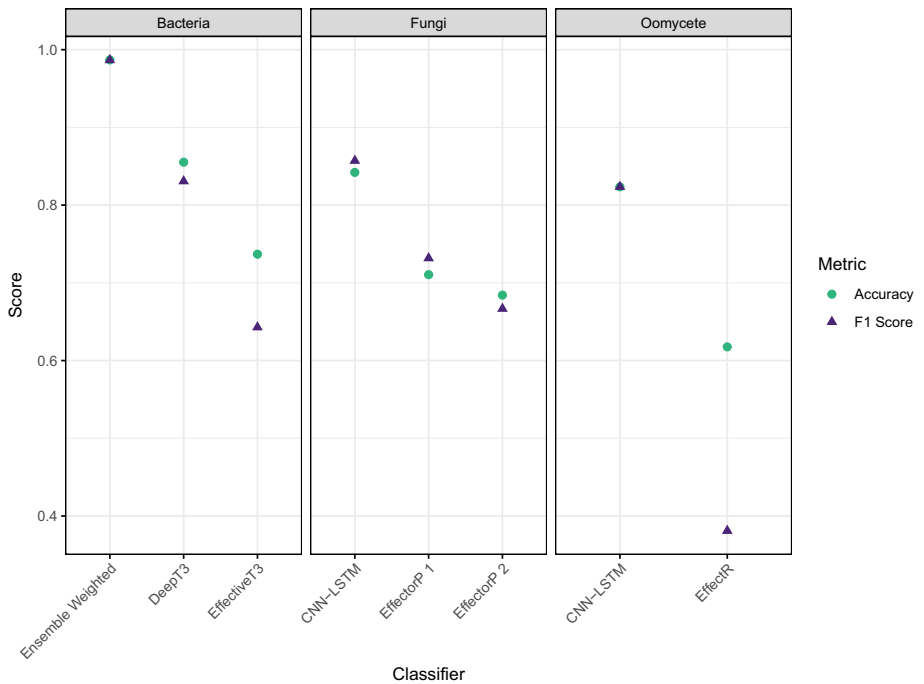

Fig. 8 Accuracy and F1 score of the classifications made by our models compared with those made by widely used software

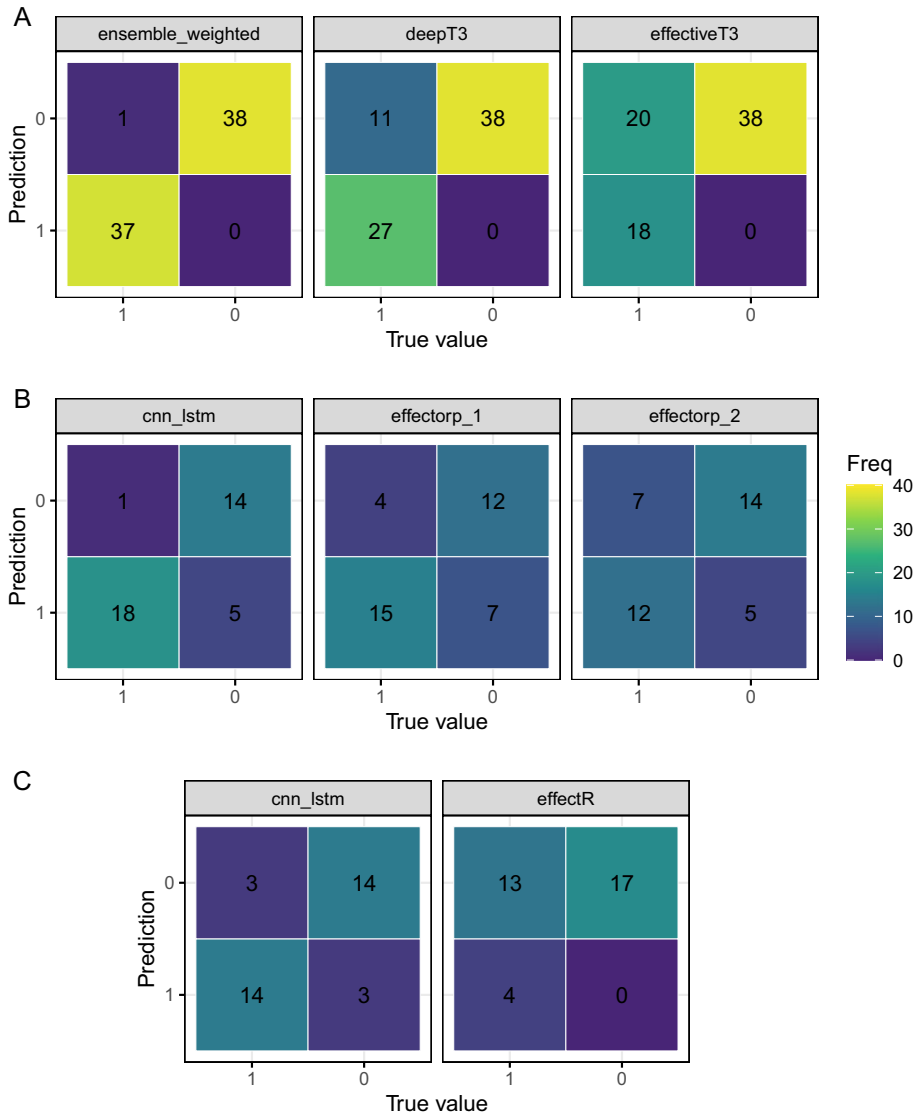

Fig. 9 Confusion matrices of classifications made by our models compared with those made by widely used software. **A** Tools evaluated on bacterial sequences; **B** Tools evaluated on fungal sequences; **C** Tools evaluated on oomycete sequences

similar scores across the training and validation set indicating a greater generability and equivalent power. Only six effector sequences are provided as examples with the oomycete specific effector finder effectR. As this is not a trainable model in the same sense as the others, no large training set is needed. We attempted classification with these and effectR was able to classify 5 of the 6, whereas our CNN-LSTM model for oomycetes classified 3 of the 6 (Fig. 10C).

Overall, our models performed more strongly than the previously available ones tested across the range of sequences examined.

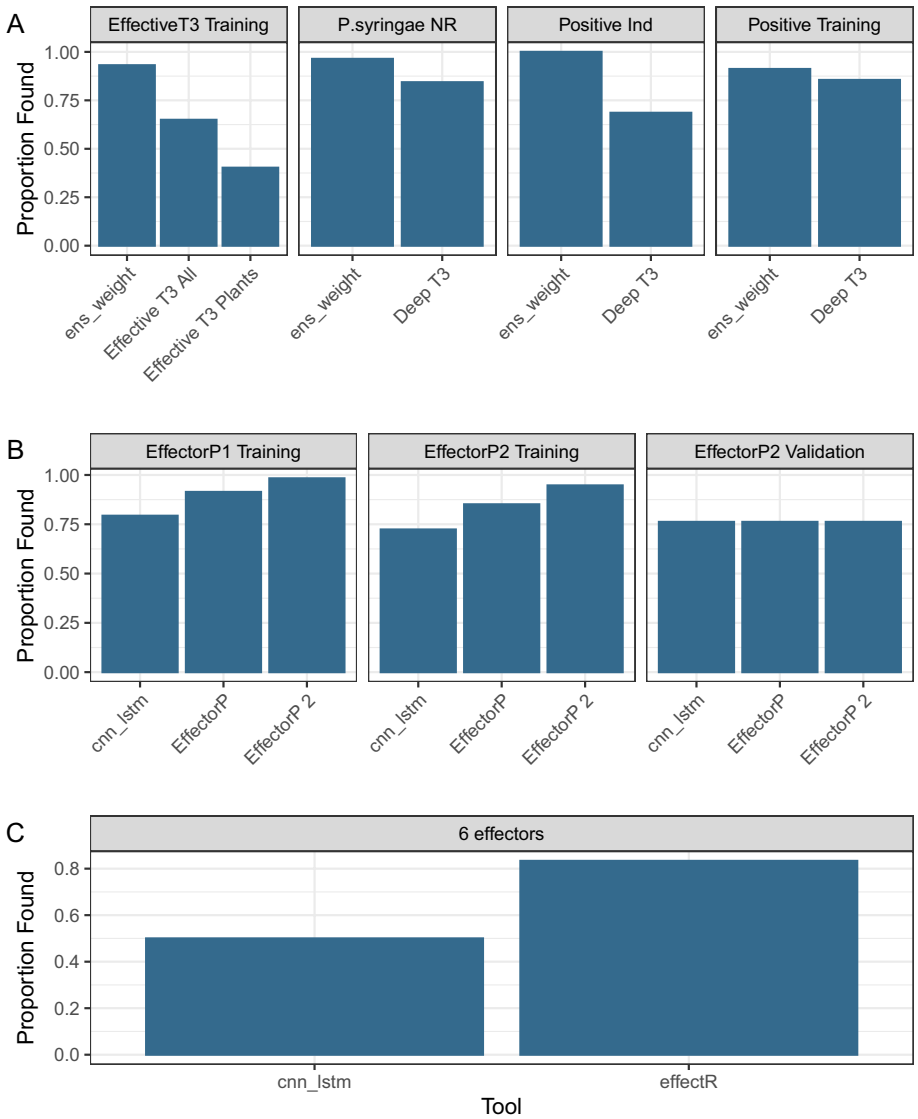

Fig. 10 Performance of models on training and validation datasets used to train other effector prediction tools. **A** Tools trained on bacterial sequences; **B** Tools trained on fungal sequences; **C** Tools trained on oomycete sequences

Convolution heatmaps

An advantage of CNNs relative to other deep neural networks is their relative interpretability. A CNN can be analysed and activation maps extracted which highlight the regions of the input to which the CNN is most strongly relying on to classify. To examine the responses of the models, we ran all sequence data sets back through the CNN-LSTM models for each taxon and extracted the network activations using the GRAD-CAM method. The profiles were smoothed using FFT and examined, (see Fig. 11). All the models showed a peak of activation at the N-terminus of the sequences, coincident with expected positions of secretion signals. The fungal sequences created a single broad activation region with a width of 50 to 100 amino acids while the bacterial and oomycete sequences create a some smaller grouped peaks in a broader region which were each

Table 3 Enrichment analysis of amino acid types

Taxon	Hydrophobic	Polar	Neutral	Basic	Acidic
Bacteria	0.44 : 0.40 (0.00)	0.26 : 0.28 (0.00)	0.09 : 0.09 (0.20)	0.11 : 0.13 (0.00)	0.09 : 0.09 (0.16)
Fungi	0.39 : 0.38 (0.06)	0.29 : 0.28 (0.08)	0.09 : 0.09 (0.29)	0.13 : 0.14 (0.00)	0.10 : 0.11 (0.26)
Oomycete	0.44 : 0.43 (0.01)	0.28 : 0.27 (0.00)	0.07 : 0.08 (0.00)	0.10 : 0.10 (0.07)	0.10 : 0.12 (0.00)

Proportions of each amino acid type in the activation region of All Sequences: Effector Sequences, with (Probability). Probability by hypergeometric test of the observed number of amino acid type in the region of activation in effector sequences in our compiled sequence set relative to a background of the same region in all sequences in that taxon

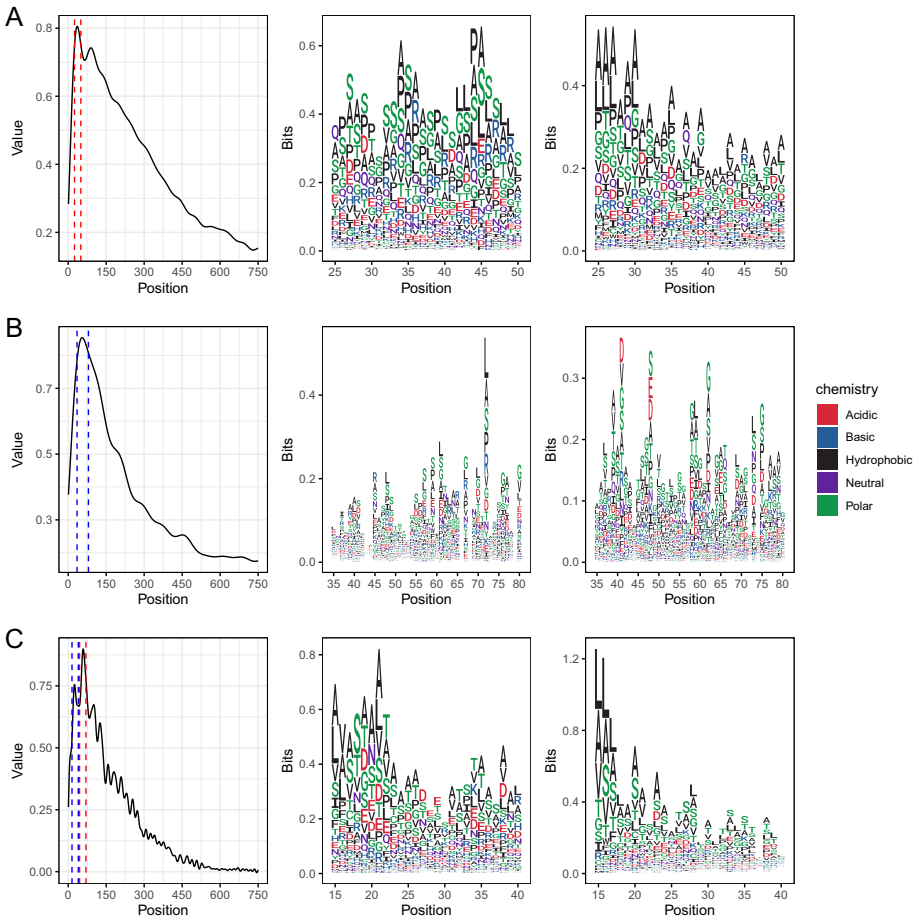

Fig. 11 Grad-CAM activation profiles of CNN-LSTM models when run with all datasets showing the region of the sequence the model is most able to use to classify (leftmost panel). The most active part of the sequences are shown in logo plots of sequences extracted from the activation region within the dotted lines for effector data (centre panel) and non-effector data (rightmost panel). **A** Models trained on bacterial sequences; **B** Models trained on fungal sequences; **C** Models trained on oomycete sequences

around 20 amino acids. We examined further the sequences under the largest peaks, specifically for bacterial sequences we used amino acids 25 to 50, for fungal sequences we used amino acids 35 to 80 and for oomycete sequences we used amino acids 15 to 40. Compositional analysis of the sequence under the peaks showed no apparent primary sequence conservation or motifs as shown in the logo plots in Fig. 11 even within a

taxon and data set. Enrichment analysis of different amino acid categories showed some statistically significant ($p < 0.05$) changes in proportions of amino acid types relative to the whole set of sequences in the activated regions (Table 3). Bacterial effectors are depleted of hydrophobic amino acids and enriched in polar and basic amino acids. Fungal effectors are enriched in basic amino acids, with no other differences. The oomycete effectors are most interesting in that their activating regions are depleted in hydrophobic and polar amino acids while enriched in neutral and acidic amino acids.

Discussion

We have compiled a set of known experimentally validated effectors from phytopathogenic bacteria, fungi and oomycetes and used them as positive training examples with which to tune a range of CNN classifiers, the sequences are all taken from the database PHI-Base, a manually curated database that aims to search all the literature on pathogen and host interactions [18]. The data in PHI-Base is complete as far as 2013 at the moment, thus the phytopathogen effectors we have collected should be all those shown experimentally to have an effect on a host that have been reported on since that time. We chose this set as we believed that this would give us the most reliable and unbiased set of effectors on which we could train learners. That is not to say that the set itself cannot be biased and that the set does not introduce any bias into the classifications of our learners. Sources of bias in our sequence set include the time limits on what has so far been included in PHI-Base, any effectors known but not reported on in this seven year time period cannot be represented in the models. The species of phytopathogens represented in the set also create bias, the effectors are not selected from species sampled proportionally or randomly but instead are those that trends in effector research over the last seven years have brought to focus. In particular, species that have had genome sequencing projects over this time are over-represented. There may also be some echoes of other methods previously applied, effectors studied experimentally must be identified first as hypothetical effectors, usually with the aid of computational tools whose models are themselves biased towards the sort of sequence that we already know. The effectors in the literature may therefore be enriched with respect to known features and results from classifiers should be interpreted with this in mind. We are not easily able to quantify any bias, but the greater generalisability of our models over the others tested gives us reason to believe that the sequence set trained on was broad enough for good models to be developed.

A common misconception of deep learning models is that training datasets need to be extremely large, this is only half true. In fact, in practice training data need only to be large with respect to the model size. Here we have coupled small data sets (tens or hundreds of training examples) with small models. The layers and architectures of the models presented here are much smaller than those typically used in large machine vision projects for example. Yet the small models remain useful and have predictive power, indicating a definite role for deep learning approaches in the work of bioinformaticians without truly massive datasets.

Training the models proved to be computationally expensive, the architectures used have a large number of parameters and hyperparameters to be optimised (see Additional

file 1: Table S4 for a breakdown) and although only a small fraction of the possible hyperparameter and parameter space was explored we compared 3000 models, at a run time of around 144 minutes for CNN-LSTM, 57.2 minutes for CNN-GRU, and 45 minutes for both GRU Embedding and LSTM Embedding. For these relatively small data sets, a significant amount of specialised GPU compute power was required.

The models we created performed exceptionally on the PHI-Base hold-out validation data set of phytopathogenic effector sequences. The greater than 80% classification accuracy for the fungal and oomycete models is an excellent accuracy on such sequences and our models outperformed the other classifiers tested by large margins on the PHI-Base fungal, no other machine learning method has been reported to have performed as well on phytopathogen sequences. The greater than 97% accuracy we observed in our model trained on bacterial phytopathogen effectors is also exceptional and similar to what Xue et al. [17] showed in human bacterial pathogen effectors. When we evaluated the proportion of effectors each of our and other classifiers could find in data used to create the other classifiers, we found that our bacterial model outperformed the bacterial models again. A slightly different picture emerged when we compared our method with EffectorP 1 and 2 on fungal data. Both versions of Effector P outperformed our model on training data used to create Effector P in the first instance but identical predictions were made by all fungal classifiers on the validation set provided by Effector P 2. This was coincident with a large drop in accuracy from training to validation data by Effector P 1 and 2. Combined with observations on our PHI-Base data on higher numbers of false positives from Effector P, we conclude that Effector P is over-fitted slightly on its training data and that our model is likely to be more generally accurate. We expect that the tools used in concert will provide very good classification and prediction of fungal effectors. Our model also performed worse on the data used to test the untrained heuristic oomycete RXLR sequence detector EffectR, finding three of six, relative to EffectR's five of six. This is likely due to the six test sequences being exclusively RXLR and our model being trained on effectors in general, the presence of RXLR not being diagnostic of effectorness in our model means Researchers hoping to find exclusively RXLR containing proteins would be advised to use EffectR, those hoping to find effectors beyond RXLRs may find utility with our oomycete model.

In developing a negative training data set that contained secreted non-effector proteins we hoped to decouple the identification of effector from identification of secreted protein as a proxy for effector. We believe that we have been succesful at this, the models we developed do identify effectors against a background of secreted proteins, indicating that they have some internal representation of what makes an effector different from a secreted peptide. By examining the activation maps of the CNN models, we learned that the maximum activations in the models remains in the N-terminus of the proteins, coincident with the expected positions of secretions signals and is relatively narrow (about 25 amino acids. We also noted that there is no typical primary sequence motif that can be identified, the identity of the amino acids themselves does not seem important. We did find that various categories of amino acid were enriched or depleted significantly in the narrow activation region of effectors, relative to the same area of non-effector secreted proteins. A favouring of functional properties of the amino acids of the effector over

the actual identity may be reflective of the many functional types origins of effectors. It may be that the N-terminal activation region in the effectors represents non-canonical secretion-compatible signal in the effector proteins. In order to evolve towards a useful secretion signal from an arbitrary point it may be enough to have a region that satisfies some chemical property that makes it secretable, yet is different enough in effectors to separate them from the secreted proteins by a deep learning algorithm.

Finally, we have made our models available in an R package *deepredeff* - from 'deep learning prediction of effector'. This can be obtained from GitHub as shown in methods, installation and usage instructions are available in the documentation provided there. The R package allows the user to run the sequences from various sequence format files against a selected model and obtain a *p* value that each sequence is an effector according to the model. Various summaries and plots are provided for the user when working on large numbers of sequences. The package integrates well with Bioconductor. For those wishing to use Python to make use of our models, we provide the models as HDF5 files from TensorFlow that can be use in that toolkit.

Conclusions

We used ensemble deep learning models based on convolutional neural networks to predict effectors of any class in bacteria, fungi and oomycetes more accurately than any previous method. The power of the models in classification comes from a highly-accurate curated dataset of positive effectors and the ability of the neural networks to work without feature selection. By not selecting which features we believe would be useful to classify upon we have relieved the models from being limited by potentially erroneous suppositions by the experimenter. Doing so allowed the models to find patterns in the protein sequence that are related to the state of being an effector and classify using them. Although the patterns could not be clearly extracted from the internal representations of the models our analysis showed that the classifiable segments were primarily in the N-terminal 25 most residues of the effector proteins. The resulting tool we developed to allow use of the models *deepredeff* currently outperforms the most widely used effector prediction tools on the high quality training data we provided and the data that was used to train and assess those other tools.

Availability and requirements

Project name: deepredeff
Project home page: https://ruthkr.github.io/deepredeff
Operating system(s): Platform independent
Programming language: R
Other requirements: None
License: MIT
Any restrictions to use by non-academics: None

Abbreviations
ML: Machine learning; CNN: Convolutional neural network; MPMI: Molecular plant microbe interactions; LSTM: Long short term memory; GRU: Gated recurrent unit; CPU: Central processing unit; GPU: Graphics processing unit; TM: Trademark; TP: True positive; FP: False positive; FN: False negative; TN: True negative; Grad-CAM: Gradient-weighted class activation mapping; ID: Identifier; FFT: Fast fourier transform.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04293-3.

Additional file 1. Supplemental Tables.

Acknowledgements
We wish to thank Martin Page, Ram Krishna Shrestha, the NBI Computing Infrastucture for Science Team for computational support and Karen Smith for researcher support. We also to wish the staff of PHI-Base, particularly Dr Martin Urban for assistance in querying and bulkdownload from their database.

Authors' contributions
R.K. carried out experiments and developed analysis code, and developed the software tools. D.M. designed experiments, interpreted results, and wrote the manuscript. Both authors have read and approved the manuscript.

Funding
RK and DM were supported by The Gatsby Charitable Foundation core grant to The Sainsbury Laboratory. The funding body did not play any role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials
The datasets generated and/or analysed during the current study are available in the 'ruth-effectors-prediction' repository, https://github.com/TeamMacLean/ruth-effectors-prediction. Individual datasets location within this repository are listed per dataset in the Results section. The R package created is available at https://ruthkr.github.io/deepredeff.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
We have no competing interests.

Received: 24 March 2021 Accepted: 8 July 2021
Published online: 17 July 2021

References

1. Nalley L, Tsiboe F, Durand-Morat A, Shew A, Thoma G. Economic and environmental impact of rice blast pathogen (*Magnaporthe oryzae*) alleviation in the United States. PLoS ONE. 2016;11(12):0167295. https://doi.org/10.1371/journal.pone.0167295.
2. Haas BJ, Kamoun S, Zody MC, Jiang RH, Handsaker RE, Cano LM, Grabherr M, Kodira CD, Raffaele S, Torto-Alalibo T, et al. Genome sequence and analysis of the Irish potato famine pathogen Phytophthora infestans. Nature. 2009;461(7262):393–8. https://doi.org/10.1038/nature08358.
3. Nion YA, Toyota K. Recent trends in control methods for bacterial wilt diseases caused by Ralstonia solanacearum. Microbes Environ. 2015;30(1):1–11. https://doi.org/10.1264/jsme2.ME14144.
4. Franceschetti M, Maqbool A, Jiménez-Dalmaroni MJ, Pennington HG, Kamoun S, Banfield MJ. Effectors of filamentous plant pathogens: commonalities amid diversity. Microbiol Mol Biol Rev MMBR. 2017;81(2):00066–16. https://doi.org/10.1128/MMBR.00066-16.
5. Sperschneider J, Dodds PN, Gardiner DM, Manners JM, Singh KB, Taylor JM. Advances and challenges in computational prediction of effectors from plant pathogenic fungi. PLoS Pathog. 2015;11(5):1004806. https://doi.org/10.1371/journal.ppat.1004806.
6. Savojardo C, Martelli PL, Fariselli P, Casadio R. DeepSig: deep learning improves signal peptide detection in proteins. Bioinformatics (Oxford, England). 2017;33:831. https://doi.org/10.1093/bioinformatics/btx818.
7. Sperschneider J, Dodds PN, Singh KB, Taylor JM. ApoplastP: prediction of effectors and plant proteins in the apoplast using machine learning. New Phytol. 2018;217(4):1764–78. https://doi.org/10.1111/nph.14946.
8. Sperschneider J, Dodds PN, Gardiner DM, Singh KB, Taylor JM. Improved prediction of fungal effector proteins from secretomes with effectorp 2.0. Mol Plant Pathol. 2018;19(9):2094–110. https://doi.org/10.1111/mpp.12682.

9. Sperschneider J, Gardiner DM, Dodds PN, Tini F, Covarelli L, Singh KB, Manners JM, Taylor JM. Effectorp: predicting fungal effector proteins from secretomes using machine learning. New Phytol. 2016;210(2):743–61. https://doi.org/10.1111/nph.13794.
10. Jurtz VI, Johansen AR, Nielsen M, Almagro Armenteros JJ, Nielsen H, Sønderby CK, Winther O, Sønderby SK. An introduction to deep learning on biological sequence data: examples and solutions. Bioinformatics (Oxford, England). 2017;33(22):3685–90. https://doi.org/10.1093/bioinformatics/btx531.
11. Lawrence S, Giles CL, Back AD. Face recognition: a convolutional neural-network approach. IEEE Trans Neural Netw. 1997;8(1):98–113. https://doi.org/10.1109/72.554195.
12. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in neural information processing systems, vol. 25. Red Hook: Curran Associates, Inc; 2012. p. 1097–105.
13. Pyrkov TV, Slipensky K, Barg M, Kondrashin A, Zhurov B, Zenin A, Pyatnitskiy M, Menshikov L, Markov S, Fedichev PO. Extracting biological age from biomedical data via deep learning: too much of a good thing? Sci Rep. 2018;8(1):5210. https://doi.org/10.1038/s41598-018-23534-9.
14. Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference on machine learning. ICML '08. New York, NY, USA: ACM; 2008. p. 160–7. https://doi.org/10.1145/1390156.1390177.
15. Wallach I, Dzamba M, Heifets A. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. CoRR arXiv:http://arxiv.org/abs/1510.02855. (2015)
16. MacLean D. A convolutonal neural network for predicting transcriptional regulators of genes in arabidopsis transcriptome data reveals classification based on positive regulatory interactions. bioRxiv. 2019. https://doi.org/10.1101/618926.
17. Xue L, Tang B, Chen W, Luo J. DeepT3: deep convolutional neural networks accurately identify Gram-negative bacterial type III secreted effectors using the N-terminal sequence. Bioinformatics. 2018;35(12):2051–7. https://doi.org/10.1093/bioinformatics/bty931.
18. Urban M, Cuzick A, Seager J, Wood V, Rutherford K, Venkatesh SY, De Silva N, Martinez MC, Pedro H, Yates AD, Hassani-Pak K, Hammond-Kosack KE. PHI-base: the pathogen-host interactions database. Nucleic Acids Res. 2019;48(D1):613–20. https://doi.org/10.1093/nar/gkz904.
19. Yates AD, Achuthan P, Akanni W, Allen J, Allen J, Alvarez-Jarreta J, Amode MR, Armean IM, Azov AG, Bennett R, Bhai J, Billis K, Boddu S, Marugán JC, Cummins C, Davidson C, Dodiya K, Fatima R, Gall A, Giron CG, Gil L, Grego T, Haggerty L, Haskell E, Hourlier T, Izuogu OG, Janacek SH, Juettemann T, Kay M, Lavidas I, Le T, Lemos D, Martinez JG, Maurel T, McDowall M, McMahon A, Mohanan S, Moore B, Nuhn M, Oheh DN, Parker A, Parton A, Patricio M, Sakthivel MP, Abdul Salam AI, Schmitt BM, Schuilenburg H, Sheppard D, Sycheva M, Szuba M, Taylor K, Thormann A, Threadgold G, Vullo A, Walts B, Winterbottom A, Zadissa A, Chakiachvili M, Flint B, Frankish A, Hunt SE, IIsley G, Kostadima M, Langridge N, Loveland JE, Martin FJ, Morales J, Mudge JM, Muffato M, Perry E, Ruffier M, Trevanion SJ, Cunningham F, Howe KL, Zerbino DR, Flicek P. Ensembl 2020. Nucleic Acids Res. 2020;48(D1):682–8. https://doi.org/10.1371/journal.pone.016729506 (**Accessed 2020-05-11**).
20. Smedley D, Haider S, Ballester B, Holland R, London D, Thorisson G, Kasprzyk A. BioMart—biological queries made easy. BMC Genom. 2009;10(1):22. https://doi.org/10.1186/1471-2164-10-22.
21. Dyrløv Bendtsen J, Nielsen H, von Heijne G, Brunak S. Improved prediction of signal peptides: SignalP 3.0. J Mol Biol. 2004;340(4):783–95. https://doi.org/10.1016/j.jmb.2004.05.028.
22. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC Bioinform. 2009;10:421. https://doi.org/10.1186/1471-2105-10-421.
23. Van Rossum G, Drake FL Jr. Python reference manual. Amsterdam: Centrum voor Wiskunde en Informatica Amsterdam; 1995.
24. Chollet F et al. Keras. https://doi.org/10.1038/nature083580 (2015)
25. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org. 2015. http://tensorflow.org/.
26. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13(1):281–305.
27. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
28. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. Int J Comput Vis. 2019;128(2):336–59. https://doi.org/10.1007/s11263-019-01228-7 http://arxiv.org/abs/1610.02391.
29. Eichinger V, Nussbaumer T, Platzer A, Jehl M-A, Arnold R, Rattei T. EffectiveDB-updates and novel features for a better annotation of bacterial secreted proteins and Type III, IV, VI secretion systems. Nucleic Acids Res. 2015;44(D1):669–74. https://doi.org/10.1093/nar/gkv1269.
30. Tabima JF, Grünwald NJ. effectr: An expandable r package to predict candidate rxlr and crn effectors in oomycetes using motif searches. Mol Plant-Microbe Interact®. 2019;32(9):1067–76. https://doi.org/10.1094/MPMI-10-18-0279-TA.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.