Machine learning models accurately predict T-DNA insertion into plant genomes

Sawyer Smith[1] and Azing Agah[1#]

[1]Park University
[#]Advisor

ABSTRACT

Agrobacterium tumefaciens is a gram-negative bacterium of the family Rhizobiaceae and is known for its pathogenic ability to induce a neoplastic response in over 100 different species of plants, often leading to significant decline in individual plant health. The mechanism by which tumors are induced includes a segment of DNA contained within the bacterium's T_i plasmid which is integrated in the host genome. The T-DNA is oncogenic, encoding enzymes that increase the production of certain plant hormones ultimately leading to tumor formation. The impressive ability of T-DNA to integrate into plant genomes has led to its use as a common method of genetic transformation in plants. While it has been documented that the T-DNA insertion occurs at double strand breaks, the mechanism of insertion still remains elusive. Currently, the point at which the T-DNA is inserted in the host genome is believed to be somewhat random with respect to the surrounding sequences, and uncontrolled multiple insertion sites appear to be a common phenomenon. In this study, we utilized machine learning algorithms to assess the nucleotide sequences that are important in integration of Ti plasmid into the host genome. Various machine learning algorithms have yielded high-accuracy models provided the sequence data alone.

Introduction

Agrobacterium tumefaciens is a gram-negative, soil microbe and belongs to the family Rhizobiaceae and is known for its pathogenicity to induce tumor formation in a wide variety of plant species (Necrela, et. al., 2021). The bacterium contains a tumor inducing (Ti) plasmid that is essential for delivery of the pathogen into the host genome which transforms normal plant cells into autonomous tumor cells leading to crown gall disease (Gelvin 1990). The essential regions of the Ti plasmid include six major operons Vir A, VirB, Vir C, VirD, VirE and VirG encoding for virulence genes; and the transfer DNA region, a section of the Ti plasmid that is transferred via conjugation into host plant cells (Necrela, et. al., 2021). The molecular events by which Ti plasmid is integrated into the host genome still remain elusive. Agrobacterium tumefaciens is an efficient DNA delivery system for production of biomedically important macromolecules and therapeutics (Ramessur et al., 2018). Agrobacterium has been utilized to successfully engineer vaccine production in bananas and tomatoes (Gelvin, 2003). Unraveling the mechanism of DNA integration is quite appealing particularly for the developing world where access to vaccines are financially challenging.

The mechanism by which agrobacterium induces tumor formation begins with the T_i plasmid. Virulence proteins coded by Agrobacterium are involved in processing the T-DNA as well as shuttling it into the plant cell via a type-IV secretion system (Nester 2015; Zhang et al., 2017). Although the mechanism by which T-DNA is directly integrated into the host genome is not well-understood, it is plausible that virulence protein VirD2 chaperones the T-DNA to the site of insertion (Ziemienowicz, et al., 2000). Ziemienowicz et al. also demonstrated that plant extracts contain a ligase activity that leads to integration in varied sequence contexts suggesting that the host's genetic machinery is utilized to carry the integration. Because of the association between double-strand DNA breaks in host DNA and integration, it appears that the host's DNA repair machinery is involved in integration, though the exact mechanisms remain elusive (Kohler, et al., 1989, Gelvin 2021).

As the technology used to make scientific measurements becomes increasingly sophisticated, the datasets generated by these instruments become exceedingly large and complex (Nichols et al., 2019; Choi et al., 2020). Thus, new methods are needed to resolve this complexity in order to draw interpretable conclusions from said data (Janiesch et al., 2021). The term "machine learning" (ML) is used to describe a set of statistical/computational methods that can be used in order to make sense of big, seemingly messy data (Monaco et al., 2021). There are many different ML methods which concentrate on the prediction of a certain variable from a set of other variables with known values. The variable to be predicted is known as a "label", the "output", or the "y" and the variables used to predict the label are known as "features", "inputs", or "X" s (Kursh 2021).

In some datasets, the label has known values. In this case, the goal of ML is to find the relationship between the inputs and the output. This is called supervised learning and can be further broken down into two different types of tasks: classification and regression (Liu et al., 2012). Classification is used when the output can be sorted into one of multiple discrete categories (Kotsiantis 2007). In this study, we utilized classifiers in order to distinguish between sequences in which there is an adjacent insertion and ones in which there is not. Regression is used when the output is a continuous or quantitative variable. In both methods, a ML technique is used to generate an equation that relates the inputs to the known outputs in some way. After generating this equation, a new set of inputs can be plugged into the equation and an output is generated (Badillo et. al., 2020).

In this study, ML techniques were utilized to infer the labels based on similarities in features between observations, known as supervised learning. Typically, observations are sorted into groups based on their similarities and then assigned a label from a predetermined set of labels based on qualities of their group (Monaco, et. al., 2021).

Methods

Data Retrieval and Processing

Sequences flanking T-DNA insertions in *Arabidopsis thaliana* were retrieved from GenBank and were also obtained from a variety of previously performed studies (Ortega et al., 2002, Samson et al., 2002, Brunaud et al., 2002, Li et al., 2006). All of these sequences and all code used for this project are available at our Github at: https://github.com/sawyer-smith/tdna-insertion. Sequences ranged from approximately 100-800 bases long and were labeled as flanking the insertion on either the "right" or "left" border. Because these regions were sequences as part of insertional mutagenesis studies, they exclusively belong to coding regions of the Arabidopsis genome.

Sequences were exported from GenBank in the "GenBank (Full)" file format, and the resulting files were scraped for the relevant information. Due to the differences in how sequences were annotated between studies, similar but customized scraping algorithms were implemented in Python depending on how the information was organized. Essentially, the algorithms generated a Python list for each sequence containing two items: one was a string indicating whether the sequence flanked the insertion on the right or the left (i.e., "R" or "L") and the other was the sequence itself reformatted as a list, where each nucleotide is a single item.

These single sequence lists were then compiled into two larger lists. One of these lists included all of the right border sequences, and the other included all of the left border sequences (Figure 1). Sequences with no insertion were then randomly taken from the Arabidopsis genome and added to each list, with each final list containing equal numbers of insertion sequences and non-insertion sequences. It was verified that there were no repeat sequences in the final complete list of sequences. Before transferring to a Pandas DataFrame, each flanking sequence was labeled with a 1, and each non-flanking sequence was labeled with a 0 in order to facilitate classification using machine learning. The final set of left-border sequences plus non-insertion cases contained 488 sequences. The final set of right-border sequences plus non-insertion sequences contained 314 sequences (Figure 1).

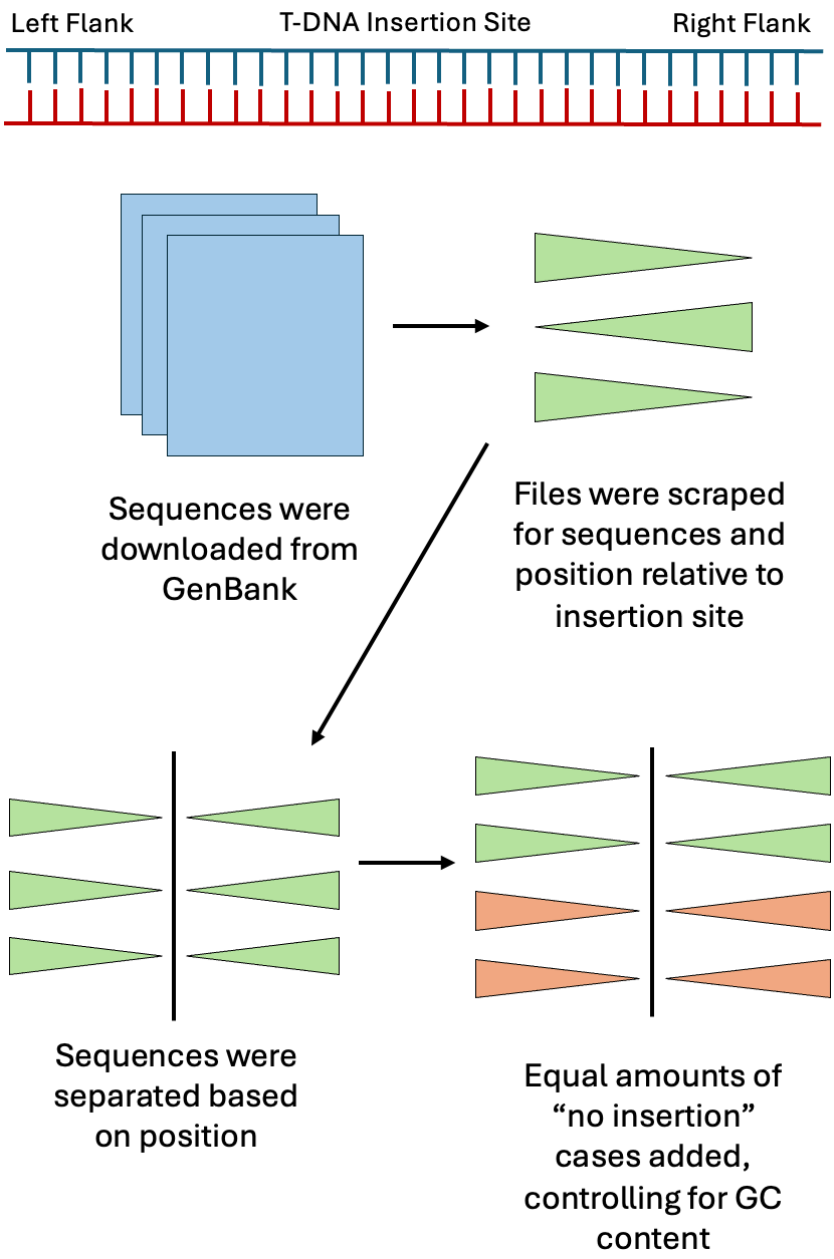

Figure 1. Sequences flanking sites of T-DNA insertion in *Arabidopsis thaliana* were retrieved from GenBank Nucleotide, originating from several studies. The sequences were then shortened and sorted into lists of right and left flanking sequences. Sequences of no insertion were taken from the *A. thaliana* genome and added to the previously generated lists. No-insertion cases were added such that the distributions of GC content between insertion and no-insertion cases were not significantly different.

The sequences that flanked insertions were all taken from regions of coding DNA, whereas the others were taken from random positions on the genome. In order to make sure that sequences were being discriminated against based on a putative quality that informed insertion, and not based on GC content, the distributions of GC content of insert and non-insert samples were measured and compared using a Mann-Whitney U test (Figure 2).

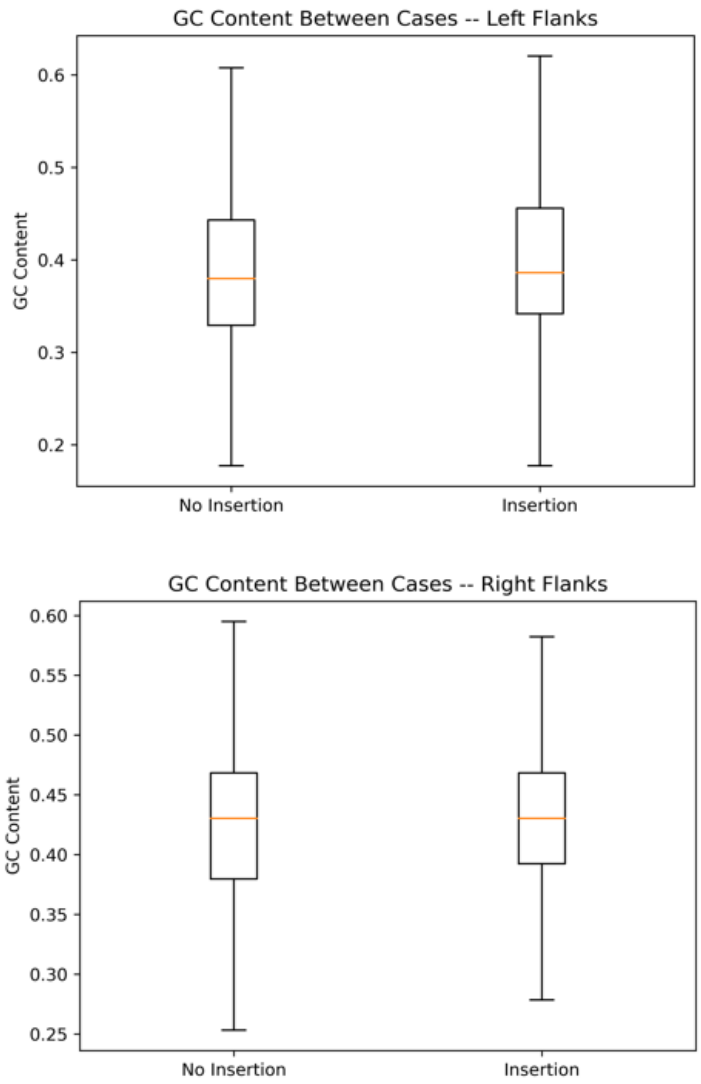

Figure 2. The distribution of GC content of insert and non-insert samples measured and compared using Mann-Whitney U Test.

One-Hot Encoding

Additionally, since machine learning models are not designed to handle qualitative data, sequences needed to be encoded before ML models could be fit to the data. One-hot encoding was used to make the data palatable for the ML models (Dahouda et al., 2021). One-hot encoding involves expanding each feature into a number of "dummy variables" based on the number of different possible values for the qualitative variable. In this case, since there are four possible values (a, g, c, and t) for each feature, each feature was expanded into four variables. Each of the four dummy variables denotes whether or not the original feature's value is a specific value out of the four possible values. Say that the first dummy variable denotes whether the original feature value is "a", the second denotes whether it is "g", the third denotes "c", and the fourth denotes "t". A one would be assigned to the dummy variable with the original

feature value, whereas zeros would be assigned to the other dummy variables. If the original feature value was “c”, for example, the respective values of the dummy variables would be 0, 0, 1, 0.

Machine Learning Models

All machine learning algorithms were implemented using libraries in Python. Various methods of binary classification were used, including logistic regression, decision trees, support vector machines (SVM), multi-layer perceptron (MLP), and convolutional neural networks (CNN). The code used for each method can be found on our GitHub. Logistic regression, decision trees, SVM, and MLP were implemented using Sci-Kit learn, a popular machine learning library in python (CITE). CNN was implemented using python libraries TensorFlow and Keras. All models were optimized using Gridsearch algorithms with cross validation.

Model Evaluation

Various methods were used for model evaluation. Accuracy was often used to evaluate both training and testing of models because classes were exactly balanced. That is, there were equal amounts of insertion cases and no-insertion cases. Similarly, receiver operating characteristic curves (ROC) were used. ROC curves plot the false positive rate against the false negative rate at different decision thresholds. The area under the curve (AUC) is used as a metric of success in prediction.

Feature Importance Analysis

Feature importance analysis is a method by which one can determine which features contribute the most to the accuracy of a ML model. This is done by taking all of the data from the feature of interest and randomly permuting it, so that each sample is randomly assigned a new value for that feature. Feature importance analysis was carried out using the model with the best performance as determined by testing accuracy and the AUC when plotted on a receiver operating characteristic (ROC) curve.

Since most pre-written feature importance analysis algorithms are not designed to handle data that has been encoded using one-hot encoding, custom algorithms were used to carry out the analysis. Essentially, the algorithm establishes a baseline accuracy by training an optimized model on non-permuted, encoded training data and then finding the testing accuracy of the model. Then, the algorithm iterates through the non-encoded training data, permutes one feature at a time, encoding the training data and fitting the ML model for each feature that gets permuted. In each iteration, the trained model is then used to make predictions based on encoded testing data, and the accuracy of those predictions (known as the model’s testing accuracy) is subtracted from the baseline accuracy in order to determine the importance of the permuted feature. The train/test split is preserved for each time the model is fit so that the model is being evaluated based on the same testing data each time.

Results

Various types of ML modes can be trained to predict insertion based on left-flanking sequences

Models trained on data from sequences on the left flank of T-DNA insertions were able to achieve high training accuracies across statistical treatments (Figure 3). The support vector classifier (SVC) was particularly accurate after optimization, achieving an AUC of 0.96.

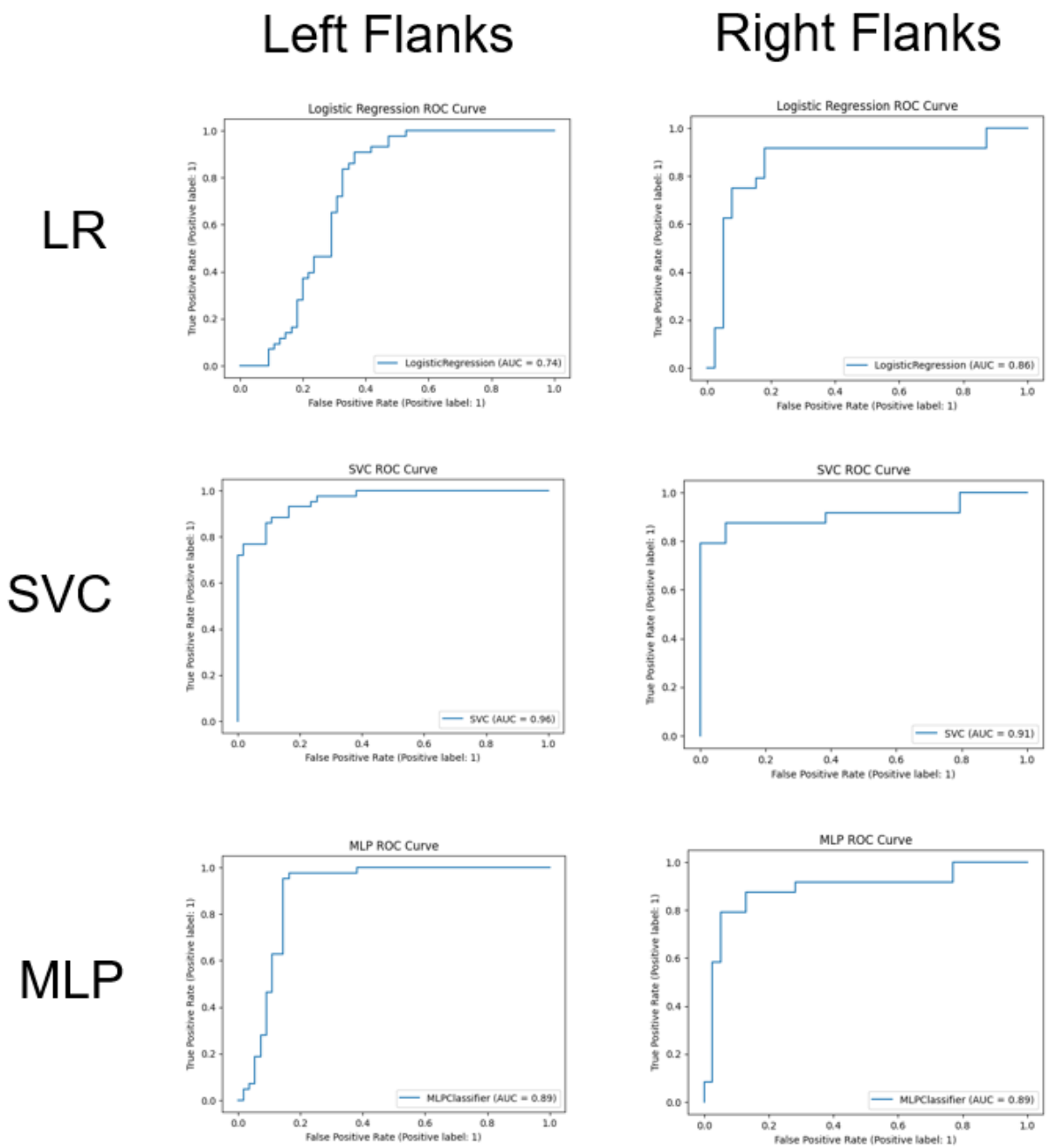

Figure 3. ROC curves were generated using testing data with previously trained ML models. The SVC models attained the greatest area under the curve, indicating high testing performance.

Various types of ML modes can be trained to predict insertion based on right-flanking sequences

The models trained on data from sequences on the right flank of T-DNA insertion also yielded high accuracies, somewhat higher even than those trained on the left flank sequences. As indicated in Figure 3, the SVC was once again the most accurate type of model trained on this data.

ML models also achieve high testing accuracies

In order to determine whether the models' performance was simply based on overfitting, testing accuracy of each model was determined. Given the previously unseen data, models yielded consistently high accuracies. As demonstrated in Figure 4, the SVC yielded the best accuracies in both models trained by left flank sequences or right flank sequences. Additionally, confusion matrices were generated for each model to determine whether models were performing poorly at predicting certain classes (insertion or no insertion) (Figure 5).

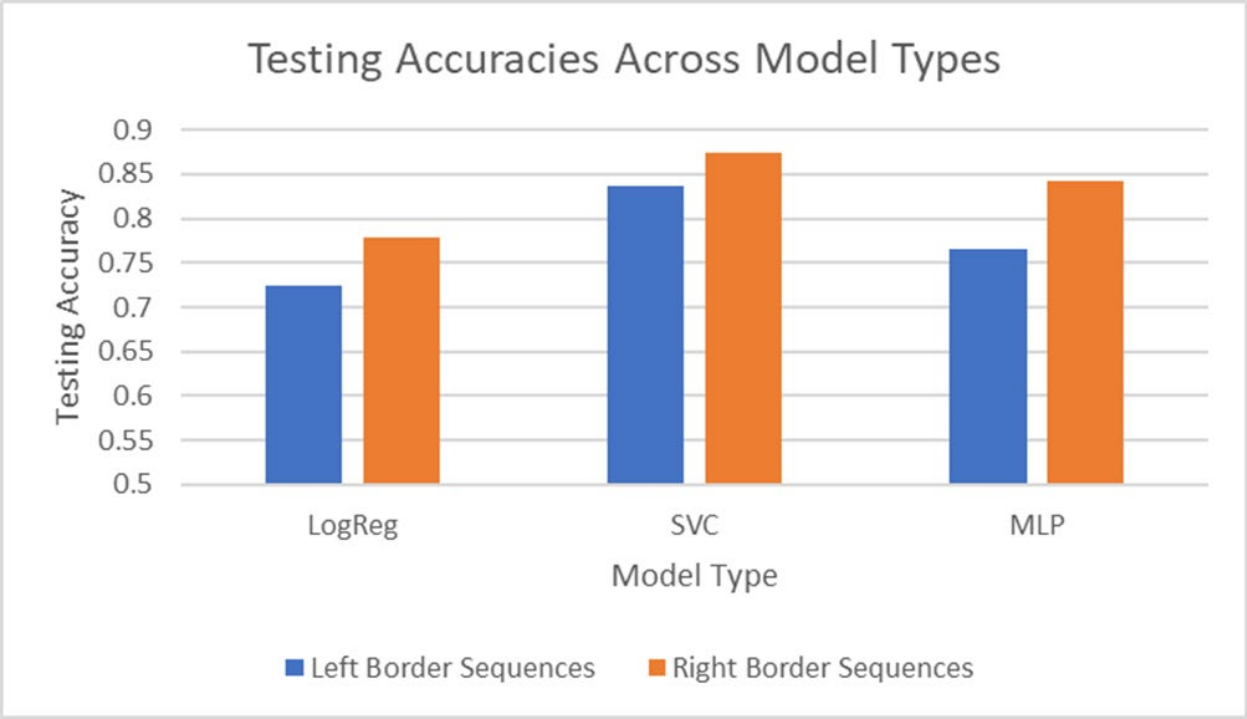

Figure 4. Comparison of accuracy between models when tested on unseen data. All three models yielded high accuracy. SVC appeared to achieve the highest testing accuracy.

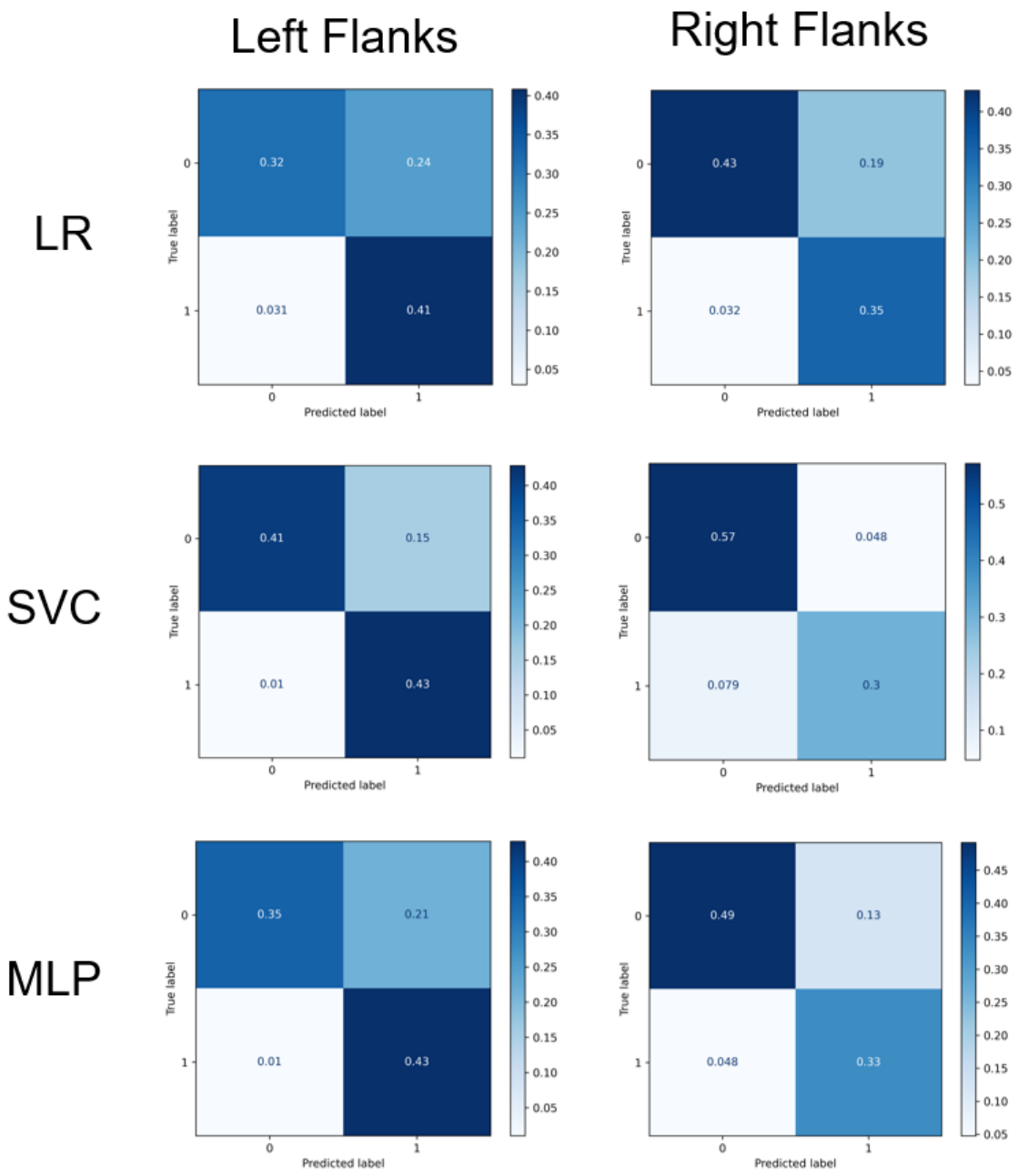

Figure 6. Confusion matrices for each of the different types of machine learning models trained on right and left flank data. Values for each of the categories were normalized to the total number of samples used for testing. In each case, the models' prediction of insertion tended to line up with the actual insertion status of the sample (true positives and true negatives were more common than false positives and false negatives).

Permutation analysis does not show any particular regions of interest

When doing permutation analysis, it was expected that there may be certain regions of the sequence where the positions in that region tended to have higher importance, or that positions closer to the insertion would have higher

importance than those that are farther away. However, the importance of each feature did not appear to follow any particular pattern in terms of their position on the sequence. This was true for the models trained on the left flanking sequences as well as those trained on the right flanking sequences (Figure 5).

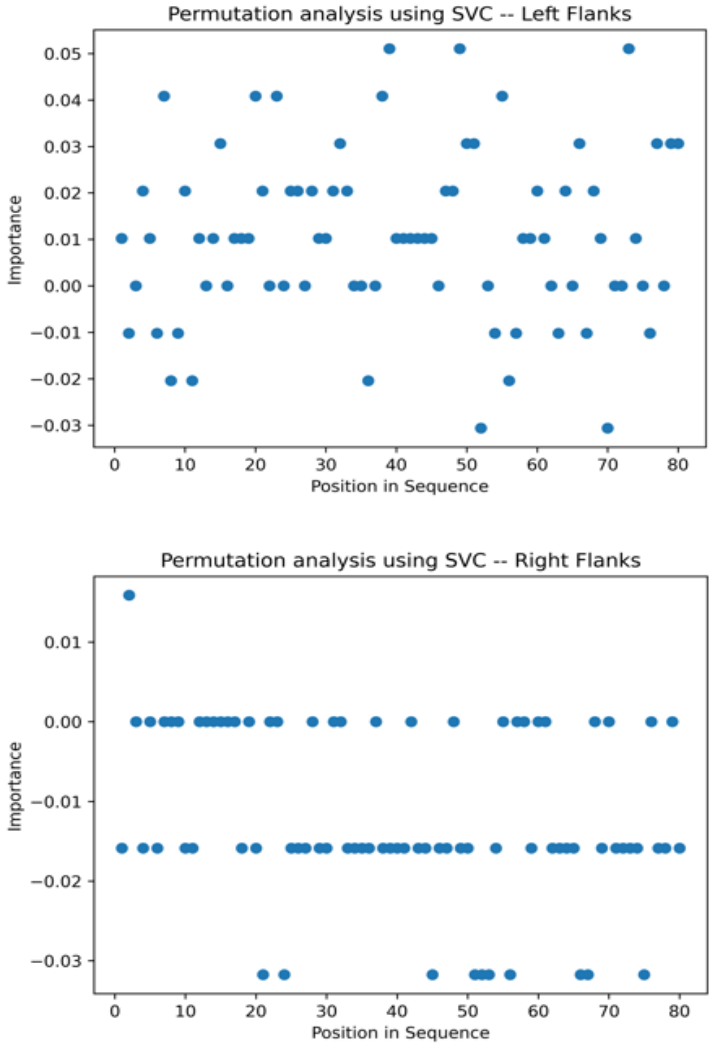

Figure 6. Figure importance analysis using the SVC models trained on the left and right flank data. The analysis did not reveal any particular regions in the flanking sequences that were more or less predictive of insertion.

Discussion

Across the board ML models performed well when predicting insertion based only on sequence data. This data was robust for predicting insertion using varied statistical treatments. This is interesting considering previous studies which have suggested that insertion is not sequence dependent. However, given the small and somewhat limited dataset, it is plausible that the models' accuracy can be somewhat attributed to overfitting, or perhaps the models were distinguishing between insertion and non-insertion cases based on properties which are not inherent to all cases of insertion. Especially concerning is the lack of insertion cases originating from non-coding DNA. One way that this possibility

was combated was by controlling for the distribution of GC content in non-insertion cases, but perhaps other qualities of non-coding DNA confounded the model-generation process.

Carrying out feature importance analysis on these data using the optimized models demonstrated that there were not any particular regions of the sequences that were especially consequential to determining whether an insertion would take place. One might expect that, for example, positions nearest to the insertion would have the most importance in determining model accuracy, but no such pattern was present in this data. This could mean that these models are overfit to this data, or that patterns determining insertion at the sequence level are more complex.

Though the model that was trained performed well for various traditional machine learning benchmarks, biological phenomena are highly complex and multifactorial. Sequence data alone may be able to explain some insertion cases but probably does not capture the full extent of features that control the probability of insertion. Therefore, there are other types of data that may be helpful in generating a more accurate and representative model. One type of data that would be particularly helpful is chromatin accessibility. It is thought that the insertion of T-DNA is connected to the presence of double-strand breaks in the host genome.

Using data from sequences flanking insertions of T-DNA as well as sequences not flanking insertions in *Arabidopsis thaliana*, accurate classifiers were generated. These classifiers not only performed well on classifying training data, but also on classifying testing data after being trained. Though these models performed well, there are certainly other types of data that could be used for training the models that may make them more generally accurate and better reflect the intricacies of T-DNA insertion.

References

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg, Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. Clinical pharmacology and therapeutics, 107(4), 871–885. DOI: 10.1002/cpt.1796

Brunaud, V., Balzergue, S., Dubreucq, B., Aubourg, S., Samson, F., Chauvin, S., Bechtold, N., Cruaud, C., Derose, R., Pelletier, G., Lepiniec, L., Caboche, M., & Lecharny, A. (2002). T-DNA integration into the Arabidopsis genome depends on sequences of pre-insertion sites. EMBO Reports, 3(12), 1152–1157. DOI: 10.1093/embo-reports/kvf237

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. C. (2020). Introduction to Machine Learning, Neural Networks, and Deep Learning. Translational Vision Science & Technology, 9(2), 14. DOI: 10.1167/tvst.9.2.14

Dahouda, M. K., & Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, *9*, 114381–114391. DOI:10.1109/ACCESS.2021.3104357

Gelvin, S. B. (1990). Crown Gall Disease and Hairy Root Disease. Plant Physiology, 92(2), 281–285. DOI: 10.1104/pp.92.2.281

Gelvin, S. B. (2003). Agrobacterium-Mediated Plant Transformation: the Biology behind the "Gene-Jockeying" Tool. Microbiology and Molecular Biology Reviews, 67(1), 16–37. DOI: 10.1128/MMBR.67.1.16-37.2003

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3), 685–695. DOI: https://doi.org/10.1007/s12525-021-00475-2

Kohler, F., Cardon, G., Pöhlman, M., Gill, R., Schieder, O. (1989). Enhancement of transformation rates in higher plants by low-dose irradiation: Are DNA repair systems involved in the incorporation of exogenous DNA into the plant genome? *Plant Mol Biol* 12, 189–199. DOI: 10.1007/BF00020504

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica (Lithuanian Academy of Sciences), 31(3), 249–268.

Kursh, S. (2021). An Introduction to the "How To" for AI and Machine Learning. Business Education Innovation Journal, 13(2), 14–23.

Lapham Rachelle A., Lan-Ying, L., Eder, X., Esteban Ganan, G., Nivya, V. M., Gelvin Stanton, B. (2021). Agrobacterium VirE2 Protein Modulates Plant Gene Expression and Mediates Transformation From Its Location Outside the Nucleus, *Frontiers in Plant*. DOI: https://doi.org/10.3389/fpls.2021.684192

Li, Y., Rosso, M. G., Ülker, B., & Weisshaar, B. (2006). Analysis of T-DNA insertion site distribution patterns in Arabidopsis thaliana reveals special features of genes without insertions. Genomics, 87(5), 645–652. DOI: https://doi.org/10.1016/j.ygeno.2005.12.010

Liu, Q., & Wu, Y. (2012). Supervised Learning. In Springer eBooks (pp. 3243–3245).

Malik, H., Chaudhary, G., & Srivastava, S. (2022). Digital transformation through advances in artificial intelligence and machine learning. Journal of Intelligent & Fuzzy Systems, 42(2), 615–622.

Monaco, A., Pantaleo, E., Amoroso, N., Lacalamita, A., Lo Giudice, C., Fonzino, A., Fosso, B., Picardi, E., Tangaro, S., Pesole, G., & Bellotti, R. (2021). A primer on machine learning techniques for genomic applications. Computational and Structural Biotechnology Journal, 19, 4345–4359. DOI: 10.1016/j.csbj.2021.07.021

Nachar, N. (2008). The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution. Tutorials in Quantitative Methods for Psychology, 4(1), 13–20. DOI: 10.20982/tqmp.04.1.p013

Necreală, M., Rotaru, A., Chifan, R., Carabet, A., & Ştef, R. (2021). Evaluation of Bacterial Pathogen Agrobacterium Tumefaciens Attack on some Apple Varieties. Research Journal of Agricultural Science, 53(4), 141–148.

Nester, Eugene W. "Agrobacterium: Nature's Genetic Engineer." Frontiers in Plant Science, vol. 5, 2015. DOI: 10.3389/fpls.2014.00730

Nichols, J. D., Chan, H., & Baker, M. J. (2019). Machine learning: applications of artificial intelligence to imaging and diagnosis. Biophysical Reviews, 11(1), 111–118. DOI: 10.1007/s12551-018-0449-9

Ortega, D., Raynal, M., Laudié, M., Llauro, C., Cooke, R., Devic, M., Genestier, S., Picard, G., Abad, P., Contard, P., Sarrobert, C., Nussaume, L., Bechtold, N., Horlow, C., Pelletier, G., & Delseny, M. (2002). Flanking sequence tags in Arabidopsis thaliana T-DNA insertion lines: a pilot study. Comptes rendus biologies, 325(7), 773–780. DOI: https://doi.org/10.1016/s1631-0691(02)01490-7

Ramessur, A. D., Bothwell, J. H. F., Maggs, C. A., Gan, S. Y., & Phang, S. (2018). Agrobacterium-mediated gene delivery and transient expression in the red macroalga Chondrus crispus. Botanica Marina, 61(5), 499–510. DOI: https://doi.org/10.1515/bot-2018-0028

Samson, F., Brunaud, V., Balzergue, S., Dubreucq, B., Lepiniec, L., Pelletier, G., Caboche, M., & Lecharny, A. (2002). FLAGdb/FST: a database of mapped flanking insertion sites (FSTs) of Arabidopsis thaliana T-DNA transformants. Nucleic Acids Research, 30(1), 94–97. DOI: 10.1093/nar/30.1.94

Van Breukelen, G. (2010). Mann–Whitney U Test. Encyclopedia of Research Design. SAGE Publications, Inc. eBooks. DOI: https://doi.org/10.4135/9781412961288

Zhang, X., Van Heusden, G. P. H., & Hooykaas, P. J. J. (2017). Virulence protein VirD5 of Agrobacterium tumefaciens binds to kinetochores in host cells via an interaction with Spt4. Proceedings of the National Academy of Sciences of the United States of America, 114(38), 10238–10243. DOI: 10.1073/pnas.1706166114

Ziemienowicz, A., Tinland, B., Bryant, J., Gloeckler, V., & Hohn, B. (2000). Plant enzymes but not Agrobacterium VirD2 mediate T-DNA ligation in vitro. *Molecular and cellular biology*, *20*(17), 6317–6322. DOI: 10.1128/MCB.20.17.6317-6322.2000