**Supplementary Information**

**PlasmidHunter: Accurate and fast prediction of plasmid sequences using gene content profile and machine learning**

Renmao Tian 1, Jizhong Zhou2, Behzad Imanian 1, 3, \*

1 Institute for Food Safety and Health, Illinois Institute of Technology, Bedford Park, IL 60501, USA

2 Institute for Environmental Genomics, Department of Microbiology and Plant Biology, University of Oklahoma, Norman, OK 73072, USA

3 Food Science and Nutrition Department, Illinois Institute of Technology, Chicago, IL 60616, USA

\* Author to whom correspondence should be addressed.

**Supplementary Table 1.** Summary of the confirmed chromosomes and plasmids for the modeling based on the annotation from NCBI database and the sizes. The sequences determined as chromosome or plasmid were included in the subsequent analaysis.

|  |  |  |  |
| --- | --- | --- | --- |
| **Annotation** | **Size** | **Determined as** | **Count** |
| chromosome | > 900 kbp | chromosome | 36454 |
| chromosome | <= 900 kbp | not sure | 918 |
| plasmid | < 600 kbp | plasmid | 40806 |
| plasmid | >= 600 kbp | not sure | 802 |
| extrachrom |  | not sure | 9 |
| genomic |  | not sure | 4 |

**Supplementary Table 2.** The evaluation of the performance of models predicting the location (chromosome or plasmid) of contigs. The evaluation was based on the model predictions on validation data using the package scikit-learn. RF = random forest; DT = decision tree; NB = naïve bayes; LR = logistic regression. Methods beginning with principal component analysis (PCA) means the modeling using PCA transformed data. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Balanced accuracy** | **Log loss** | **Recall** | **Precision** | **F score** | **ROC AUC** |
| RF | 65.4% | 65.6% | 0.64 | 32.0% | 97.8% | 48.2% | 82.2% |
| DT | 63.2% | 63.5% | 0.64 | 27.5% | 98.2% | 43.0% | 64.1% |
| **NB** | **96.0%** | **96.0%** | **1.43** | **96.4%** | **95.8%** | **96.1%** | **96.0%** |
| LR | 92.4% | 92.4% | 0.20 | 98.4% | 88.0% | 92.9% | 99.0% |
| PCA-RF | 85.6% | 85.6% | 0.33 | 82.3% | 88.3% | 85.2% | 93.9% |
| PCA-DT | 82.1% | 82.1% | 3.62 | 79.6% | 83.9% | 81.7% | 86.1% |
| PCA-NB | 63.5% | 63.7% | 12.69 | 30.1% | 91.9% | 45.4% | 75.9% |
| PCA-LR | 68.4% | 68.6% | 0.53 | 40.0% | 93.7% | 56.0% | 83.7% |
| PCA-KNN | 83.7% | 83.7% | 0.92 | 83.6% | 84.0% | 83.8% | 91.7% |

**Supplementary Table 3.** The statistics of the benchmark data. The contigs of specified lengths were randomly simulated from 3,000 plasmids and 3,000 chromosomes that were not included in the modeling. Each data set has equal number of plasmid and chromosome sequences.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bechmark data ID** | **Contig length (kbp)** | **Number of sequences \*** | **Total length (Mbp)** |
| 1 | 1 | 6000 | 6.0 |
| 2 | 5 | 5194 | 26.0 |
| 3 | 10 | 4546 | 45.5 |
| 4 | 20 | 4124 | 82.5 |
| 5 | 50 | 3070 | 153.5 |
| 6 | 100 | 1810 | 181.0 |

\* The total number of sequences were less than 6000 in some data sets because some plasmids were shorter than the required lengths and were thus not included.

**Supplementary Table 4.** The evaluation of PlasmidHunter’s performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Log loss** | **Recall** | **Precision** | **F score** | **ROC AUC** | **Time used (min)** |
| 1 | 80.1% | 80.5% | 7.2 | 97.0% | 72.0% | 82.7% | 80.5% | 3.3 |
| 5 | 89.2% | 89.3% | 3.9 | 95.5% | 84.8% | 89.8% | 89.3% | 3.6 |
| 10 | 92.0% | 92.0% | 2.9 | 94.8% | 89.7% | 92.2% | 92.0% | 4.3 |
| 20 | 94.7% | 94.7% | 1.9 | 95.0% | 94.4% | 94.7% | 94.7% | 5.6 |
| 50 | 97.1% | 97.1% | 1.1 | 96.3% | 97.8% | 97.0% | 97.1% | 7.7 |
| 100 | 97.6% | 97.6% | 0.9 | 96.6% | 98.6% | 97.6% | 97.6% | 8.5 |

**Supplementary Table 5.** The evaluation of Deeplasmid (version of Feb. 10, 2022) performance on the benchmark data with different lengths. The prediction was run on a different computer with eight processors (Intel Core i7-10510U, 1.8 GHz) because the Deeplasmid cannot be limited to use only eight processors in the other computer with 128 processors. Note that CPU mode rather than GPU mode of Deeplasmid was used in this running. The evaluation was conducted using the package scikit-learn. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Log loss** | **Recall** | **Precision** | **F score** | **ROC AUC** | **Time used (min)** |
| 5 | 65.5% | 65.5% | 1.4 | 31.7% | 98.0% | 47.9% | 86.5% | 657 |
| 10 | 67.6% | 67.6% | 1.9 | 35.9% | 98.0% | 52.6% | 89.6% | 696 |
| 20 | 74.5% | 74.5% | 2.2 | 50.1% | 97.9% | 66.3% | 92.3% | 655 |
| 50 | 86.6% | 86.6% | 1.5 | 74.4% | 98.4% | 84.7% | 95.1% | 687 |
| 100 | 93.3% | 93.3% | 0.8 | 87.7% | 98.8% | 92.9% | 97.4% | 501 |

**Supplementary Table 6.** The evaluation of PlasmidVerify (version of April 30, 2020) performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. The log loss and ROC AUC were not calculated because PlasmidVerify outputs likelihood ratios rather than probabilities of predictions. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Recall** | **Precision** | **F score** | **Time used (min)** |
| 1 | 63.5% | 63.5% | 27.5% | 98.1% | 43.0% | 9.5 |
| 5 | 78.1% | 78.1% | 58.4% | 96.2% | 72.7% | 31.4 |
| 10 | 83.5% | 83.5% | 70.0% | 96.0% | 80.9% | 47.4 |
| 20 | 88.7% | 88.7% | 80.8% | 96.0% | 87.7% | 96.6 |
| 50 | 93.1% | 93.1% | 89.8% | 96.2% | 92.9% | 185 |
| 100 | 94.8% | 94.8% | 92.3% | 97.2% | 94.7% | 202.2 |

**Supplementary Table 7.** The evaluation of PlasFlow (version 1.1.0, August 15, 2018) performance on the benchmark data with different lengths. The prediction was run on a computer with one processors (AMD EPYC 7551, 1.2 GHz) assigned to the task, because PlasmidFlow does not have multiple processing mode. Threshold of 0.5 was used for classification to eliminate unclassified prediction. The evaluation was conducted using the package scikit-learn. The log loss and ROC AUC were not calculated because PlasFlow does not output probability of predictions plasmid or chromosome. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Recall** | **Precision** | **F score** | **Time used (min)** |
| 1 | 67.4% | 67.4% | 84.9% | 62.9% | 72.3% | 0.5 |
| 5 | 83.2% | 83.2% | 89.0% | 79.7% | 84.1% | 0.5 |
| 10 | 88.6% | 88.6% | 90.4% | 87.2% | 88.8% | 0.6 |
| 20 | 91.0% | 91.0% | 89.8% | 91.9% | 90.9% | 0.6 |
| 50 | 92.8% | 92.8% | 90.0% | 95.2% | 92.6% | 0.8 |
| 100 | 93.3% | 93.3% | 89.5% | 96.9% | 93.0% | 0.7 |

**Supplementary Table 8.** The evaluation of PlasForest (version of October 14, 2021) performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. The log loss and ROC AUC were not calculated because PlasForest does not output probability of prediction. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Recall** | **Precision** | **F score** | **Time used (min)** |
| 1 | 80.6% | 80.6% | 62.6% | 97.8% | 76.3% | 3.4 |
| 5 | 78.6% | 78.6% | 58.6% | 97.7% | 73.3% | 7.7 |
| 10 | 78.8% | 78.8% | 58.2% | 98.8% | 73.3% | 11.3 |
| 20 | 79.0% | 79.0% | 58.4% | 99.3% | 73.5% | 16.8 |
| 50 | 78.0% | 78.0% | 56.2% | 99.9% | 71.9% | 25.3 |
| 100 | 74.9% | 74.9% | 50.1% | 99.6% | 66.6% | 22.4 |

**Supplementary Table 9.** The evaluation of PlasClass (version 0.1.1, November 1, 2021) performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Log loss** | **Recall** | **Precision** | **F score** | **ROC AUC** | **Time used (min)** |
| 1 | 74.6% | 74.6% | 0.5 | 76.4% | 73.7% | 75.0% | 82.0% | 0.3 |
| 5 | 82.8% | 82.8% | 0.4 | 84.2% | 81.9% | 83.0% | 90.7% | 0.4 |
| 10 | 88.3% | 88.3% | 0.3 | 89.4% | 87.6% | 88.5% | 94.8% | 0.6 |
| 20 | 91.2% | 91.2% | 0.2 | 91.3% | 91.1% | 91.2% | 96.7% | 1.1 |
| 50 | 94.2% | 94.2% | 0.2 | 95.4% | 93.2% | 94.3% | 98.2% | 1.9 |
| 100 | 94.8% | 94.8% | 0.6 | 91.8% | 97.5% | 94.6% | 98.3% | 2.6 |

**Supplementary Table 10.** The evaluation of Platon (version 1.7) performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. The log loss and ROC AUC were not calculated because Platon does not output probability of prediction. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

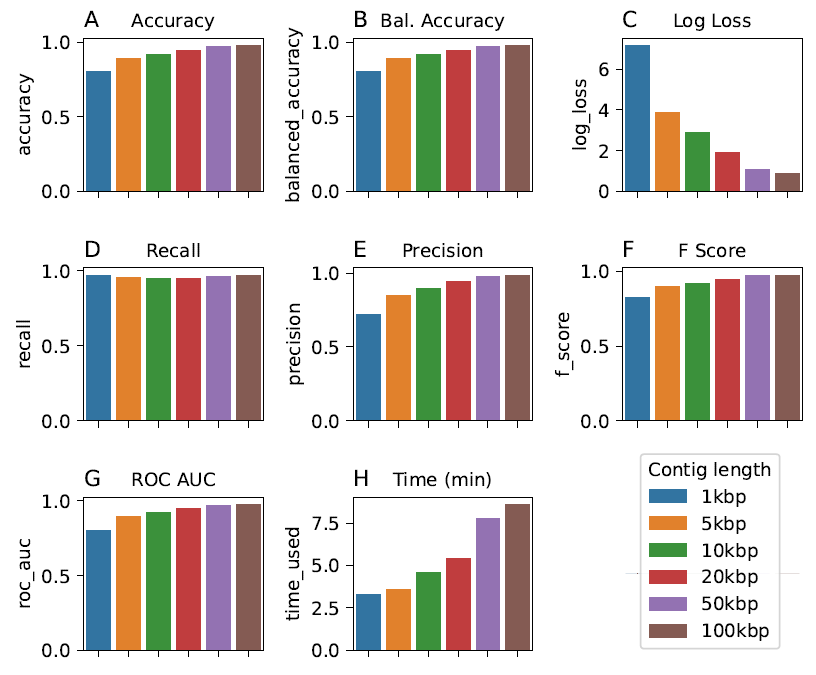
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Recall** | **Precision** | **F score** | **Time used (min)** |
| 1 | 56.1% | 56.1% | 65.3% | 55.2% | 59.8% | 47.2 |
| 5 | 65.5% | 65.5% | 83.0% | 61.5% | 70.7% | 69 |
| 10 | 65.0% | 65.0% | 82.6% | 61.1% | 70.2% | 84.5 |
| 20 | 65.6% | 65.6% | 83.6% | 61.5% | 70.9% | 30.6 |
| 50 | 66.3% | 66.3% | 85.5% | 61.8% | 71.7% | 30.4 |
| 100 | 67.6% | 67.6% | 87.5% | 62.6% | 73.0% | 33.7 |

**Supplementary Table 11.** The evaluation of MOB\_suite (version 3.1.7) performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. The log loss and ROC AUC were not calculated because MOB\_suite does not output probability of prediction. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

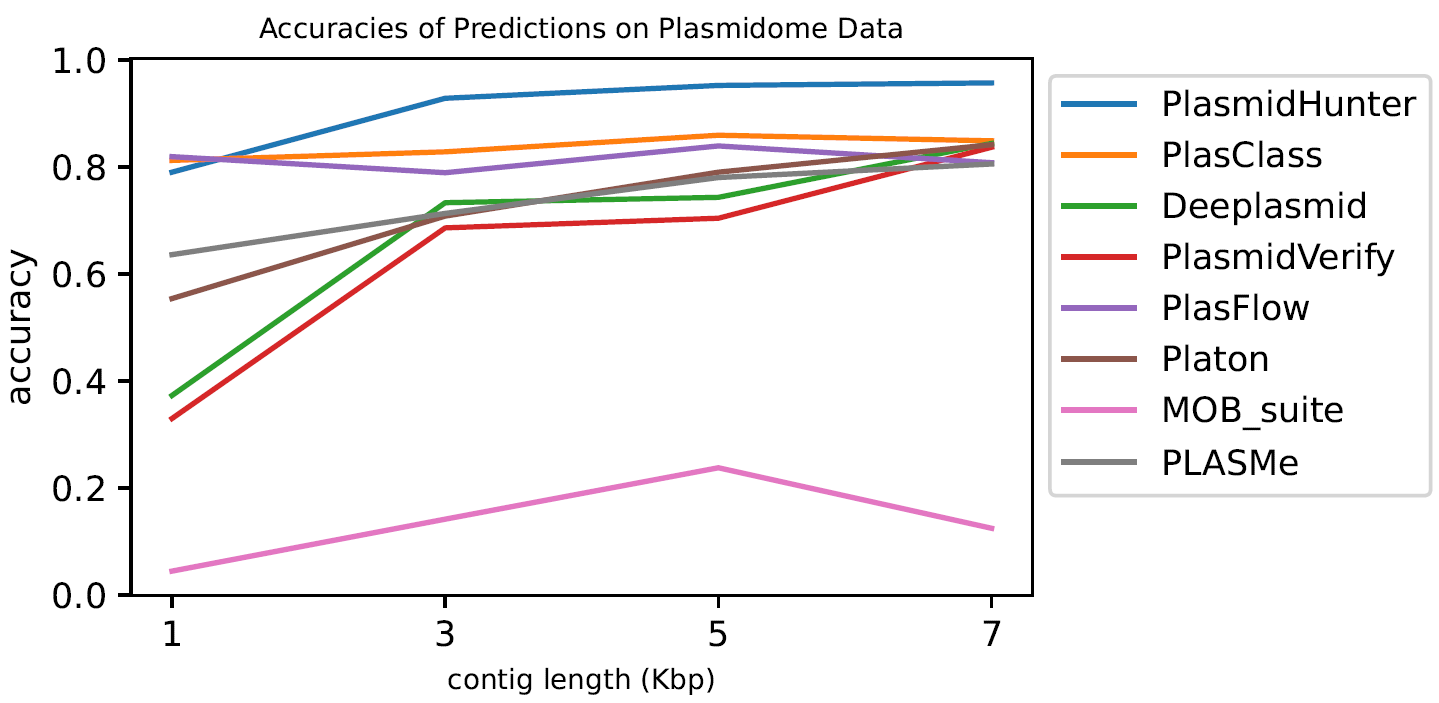
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Recall** | **Precision** | **F score** | **Time used (min)** |
| 1 | 66.8% | 66.8% | 33.8% | 99.3% | 50.5% | 22.4 |
| 5 | 77.5% | 77.5% | 55.4% | 99.2% | 71.1% | 33.8 |
| 10 | 80.9% | 80.9% | 62.2% | 99.5% | 76.5% | 33.7 |
| 20 | 80.6% | 80.6% | 61.3% | 99.6% | 75.9% | 40.4 |
| 50 | 71.2% | 71.2% | 42.6% | 99.4% | 59.6% | 56.4 |
| 100 | 66.7% | 66.7% | 33.4% | 100.0% | 50.0% | 52.6 |

**Supplementary Table 12.** The evaluation of PLASMe (version 1.1) performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. Recall was calculated as TP/(TP+FN) and Precision was calculated as TP/(TP+FP), where TP is true positive, FN is false negative, and FP is false positive. F1 Score was calculated as 2\*(Recall \* Precision) / (Recall + Precision).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contig length (kbp)** | **Accuracy** | **Balanced accuracy** | **Log loss** | **Recall** | **Precision** | **F score** | **ROC AUC** | **Time used (min)** |
| 1 | 68.5% | 68.5% | 10.3 | 89.0% | 63.2% | 73.9% | 49.5% | 6.5 |
| 5 | 70.9% | 70.9% | 9.7 | 93.8% | 64.3% | 76.3% | 50.5% | 18.6 |
| 10 | 70.9% | 70.9% | 9.8 | 94.4% | 64.2% | 76.4% | 50.1% | 28.8 |
| 20 | 71.2% | 71.2% | 9.7 | 95.4% | 64.3% | 76.8% | 50.6% | 47.3 |
| 50 | 71.2% | 71.2% | 9.5 | 96.2% | 64.1% | 77.0% | 51.1% | 81.2 |
| 100 | 69.6% | 69.6% | 9.6 | 95.1% | 63.0% | 75.8% | 51.0% | 99 |



**Supplementary Figure 1.** The visualized evaluation of PlasmidHunter’s performance on the benchmark data with different lengths. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn.

****

**Supplementary Figure 2.** The performance evaluation of all the tools visualized on the plasmidome data with different lengths. The x-axis indicates the groups of 1 - 3 kbp, 3 -5 kbp, 5 -7 kbp, 7 -9 kbp. The prediction was run on a computer with eight processors (AMD EPYC 7551, 1.2 GHz) assigned to the task. The evaluation was conducted using the package scikit-learn. The tool PlasForest was not included because of abnormal prediction results.