Benchmarking Machine Learning Methods for Eukaryote/Prokaryote Contigs Classification

Massimo Bourquin, Anita Dürr, Natasa Krčo

December 17, 2020

Abstract

An important step in the metagenomic pipeline is classifying assembled DNA sequences into Prokaryote and Eukaryote groups. For complicated samples yielding fragmented metagenomes, such as the glacier-fed streams ones, the current method fails to assign eukaryotic sequences. Our goal in this project is to improve on this method by constructing a classifier with higher accuracy. Using data transformations and optimising multiple algorithms, we are able to reach a higher accuracy compared to the current method on our training set. Particularly, K-means clustering on the features and CLR transformation of the data can improve the accuracy while reducing prediction time. If these findings can be reproduced in other k-mer based classification problems, the framework we developed could improve many other methods.

1 Introduction

In the past few years, the advent of metagenomics has had an important impact on biology, helping to understand many biological problems, e.g. disease prediction based on gut microbiome data, drinking water quality assessment or even environmental research focused on understanding ecosystems. In the standard metagenomic pipeline, a key step is to identify contigs –DNA sequences assembled based on overlap– of certain groups of organisms, allowing in the end to retrieve Metagenome Assembled Genomes, and thus to link functional discoveries with taxonomy. This step relies on the classification of the assembled DNA sequence (contigs) into the prokaryote and eukaryote groups.

The Vanishing Glaciers Project at EPFL aims to create a global-scale census of life inhabiting glacier-fed streams using metagenomics. Visual cues and laboratory assays hint for the presence of eukaryotic genomes in the samples. However, the current method for detecting eukaryotic contigs fails to assign enough of them. Our project is to build a robust and efficient classifier, i.e. that assigns each contig to Eukaryote or Prokaryote with high accuracy and low prediction time, that will be used by the NOMIS team to detect eukaryotic metagenomes in glacier-fed streams samples.

West et al. proposed in [11] in 2018 a method called EukRep based on 5-mer frequencies and a custom database. The idea is to transform DNA sequences into subsequences of length k called k-mers[1]. For each sequence, the frequency of occurrence of each possible k-mer is computed. During the past few years, the use of k-mers has gained importance, thanks to its ability to compare sequences without having to compute an alignment, particularly interesting when there is no homology between the sequences of interest. Nowadays, it is widely used in many bioinformatic applications from regulatory sequence prediction to alignment-free distance estimation, with classifiers based on linear or kernel support vector machines ([5], [11], [1]).

The method proposed in [11] reports an accuracy of more than 97.5% on a custom database and was successfully applied to environmental metagenomes (with the LinearSVC method implemented in the *scikit-learn* python library). The aim of this project is to improve on this method. For this, we use a training set based on more data, apply further transformations on the features and use different classification methods. The provided training set is based on the NCBI[2] genome database containing more than 30 000 genomes, spanning a larger diversity of species than the subset of genomes used in [11] (containing 482 genomes). Furthermore, we assess not only the k-mer frequency, but also apply data transformation methods. The first one is the Centered Log-ratio Transform (CLR), widely used in compositional data analysis. Secondly, we evaluate K-mean clustering on the features. To avoid confusion due to the same parameter name of k-mer and K-mean, we write *small* k for the k-mer size and *big* K for the number of clusters in K-mean methods. Finally, we compare the original EukRep method with the following other classifiers available in the *scikit-learn* package: LinearSVC, regular SVM, LogisticRegression, RandomForest and Multi-Layer Perceptron (MLP) neural network. This allows us to benchmark the current EukRep method in terms of accuracy and prediction time. Improvements would allow to retrieve more assembled eukaryotic sequences in the complicated metagenomes of GFS.

Even though the improvement of K-mean clustering of features was shown in [4], it was never applied to k-mer based methods. Previous work ([2], [9]) was already done using CLR transforms on other types of compositional data, with great improvement. However, current k-mer based methods rely only on Support Vector Machines trained on frequency data ([1], [2], [9]). Here we also show that the two data transformations that we propose can largely improve the accuracy and reduce the computing time of our

[1] For example, the sequence GTAC can be transformed into three 2-mers : GT, TA and AC ; and two 3-mers : GTA and TAC.

[2] National Center for Biotechnology Information

classifiers.

2 Methods

In this project, we use the *scikit-learn* machine learning library in order to build on the previous method by West et al. that uses its LinearSVC function. The choice of the library is also motivated by reproducibility purposes. We are using a seed of value 27 for the K-means clustering and 42 for all other uses. For all the work presented here, we are also setting the maximum number of iterations to be 10 000. The experiments presented in this project were carried out using the HPC facilities of the University of Luxembourg ([10]) on a 1tb memory - 128CPUs machine (Intel(R) Xeon(R) CPU E5-4660 v4 @ 2.20GHz).

In this project, we optimise the prediction time and the accuracy of the classifiers. As the aim is to create a method that can be used by others in their metagenomic pipelines, our aim is to have a classifier that is fast to apply to another dataset, in other words, that has a low prediction time. All our measures of accuracy and timing are done by 5-fold cross validation.

We first replicate the results obtained by West et al. in [11] by studying the influence of the k-mer size on the accuracy of different *scikit-learn* ([7]) methods : LinearSVC, kernel SVM, LogisticRegression, RandomForest and the Multi-Layer Perceptron neural network. We are also comparing the EukRep classifier, i.e. a LinearSVC with $C = 100$, trained on our dataset for comparison purpose. Once the choice of k-mer size is settled, we evaluate the pertinence of different data transformations by comparing the accuracy and the prediction time of those same methods. As those two first steps are a rough optimisation of the k-mer size and of the data transformation to apply, we are only interested in the influence of both parameters on the result for a given method, but don't compare the quality of the result between two different methods. For this, we use the default (hyper)parameters of the methods. We later tune these hyperparameters in order to achieve optimal accuracies for each method that we develop. The three steps are detailed in the rest of this section.

Data Cleaning

The raw datasets contain for each of the 20000 DNA sequence of 5000 nucleotides (datapoint) the number of occurrences of each k-mer (feature). As for some genomes, ambiguous characters or missing nucleotides led to a small number of k-mer, we first remove the rows containing less than 1000 k-mers. Those datapoints are not relevant for statistical results. After removing them, our dataset contains 18356 datapoints.

2.1 Influence of k-mer sizes

Datasets with k-mer size $k = 2$ to 5 are used to compute the balanced accuracies of each method. The bigger the k-mer size is, the more complex the dataset is described, so obviously the classification should be more accurate (West et al. in [11]). However this represents a cost in computation time. So the aim here is to identify what k-mer size allows a good tradeoff between complexity and computation time. Here we replicate the results from West et al. stating that $k = 5$ is the best trade-off between accuracy and computing time.

2.2 Data Transformations

For the k-mer size chosen in the previous steps, we are comparing two different data transformations to the standard count-to-frequency transformation. The first one is CLR transform, performed with a custom function [3] on the count data. The second one is the feature-aggregation of frequency data using the K-means clustering algorithms implemented in *scikit-learn* for different K values, the number of clusters.

The CLR transforms the data point $x = [x_1, ..., x_n]$ into $\text{CLR}(x) = [\log(x_1) - \log(\text{mean}(x)), ... \log(x_n) - \log(\text{mean}(x))]$. In order to apply the log function on non-positive values, we replace them by the lowest positive value. The intuition behind the CLR is that it stretches the features, in the sense that it gives less importance to high values and more importance to average values. This might enhance the description of each DNA sequence.

In a dataset of k-mer size k we have 4^k features. To reduce this large number of features we aggregate them by using K-means clustering on the features, aggregating the frequencies of the features that are assigned to each cluster. We use 16 different K values linearly distributed in $[0, 4^k]$.

To compare the effects of the above described transformations, we compute the balanced accuracy of each method as well as the prediction time. We can then identify the pairs of methods and transformations that improve on the frequency dataset.

2.3 Fine-tuning of each method

For each method and transformation, we tune the hyperparameters for the dataset with k-mer size $k = 5$, as it yields the best results (see Section 3.1). To this end, we perform a grid search over the parameter ranges displayed in Table 1 using the method GridSearchCV from the *scikit-learn* package. As before, we keep track of the overall balanced accuracy and the prediction time. We are also timing the computation time, as well as the accuracy for each class. Indeed, since the complexity of the DNA sequencing for the Eukaryote metagenomes differs from the Prokaryote metagenomes, it would be interesting to compare the accuracy of the estimators on each class. In the end, we are able to compare the methods we developed to the current one in terms of computing time and accuracy.

3 Results

3.1 Influence of the k-mer size

After comparing the performance of classifiers on datasets with k-mer sizes $\{2, 3, 4, 5\}$, we find that the optimal k-mer size is 5. As expected, the accuracy grows with k. Since previous work as well as [11] shows that $k = 5$ provides good enough results, we do further testing with the $k = 5$ dataset. Figure 1 shows a plot of the accuracy with respect to k-mer size.

[3] see CLR_transform() function in helpers.py

Method	Parameter 1	Parameter 2
LinearSVC	$C \in [10^{-2}, 10^{10}]$	/
KernelSVC (RBF Kernel)	$C \in [10^{-2}, 10^{2}]$	$\gamma \in [10^{-3}, 10^{2}]$
Logistic Regression	$C \in [1, 10^{3}]$	/
Neural Network (MLP)	Node number in the hidden layer $\in [50, 500]$	/
Random Forests	nb_trees $\in \{20, 80, 100, 150, 200\}$	depth $\in \{5, 10, 15, 20, 35, 50\}$

Table 1: Methods and parameters to tune. We performed preliminary work to find ranges that showed optima. Those are the one presented here. The parameter C in the LinearSVC, the SVM and the LogisticRegression is the inverse of the penalisation parameter lambda : the bigger the C is, the less the penalization weights in the objective function. The γ of the SVM method is the kernel coefficient. It also showed that increasing the hidden layer number only reduced the accuracy, so we tested Neural Networks with only 1 hidden layer. We decided for random forests to focus only on the size of the forest and the maximum depth of each tree. In fact, the more trees the random forest has, the more averaging the result is, but for a bigger time cost. On the other hand, having trees with too much depth can risk an overfitting of the training set.

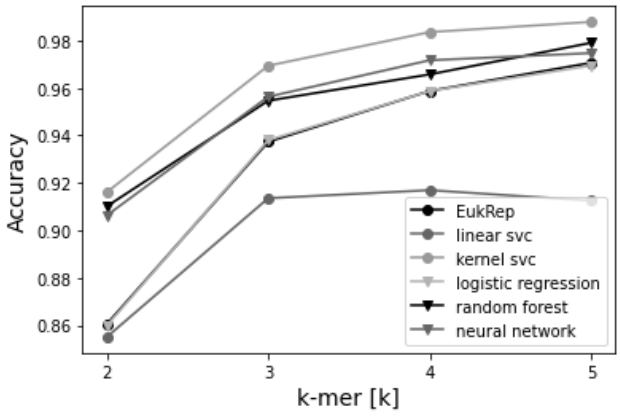

Figure 1: Mean accuracy (5-fold CV) of the different methods tested compared to the k-mer size. We can see that almost all methods show increased accuracy with bigger k-mer size, except the LinearSVC. This is due to the default parameters of this method that has hard penalization. With reduced penalization, the accuracy is much higher and the k-mer size of 5 yields more accurate predictions (see Fig. 3).

3.2 Data Transformations

For both data transformations, we find that they improve both accuracy and prediction time. K-means produces the best results for Logistic Regression ($K = 640$), KernelSVC ($K = 832$) and LinearSVC ($K = 768$), while CLR is the best for Neural Networks. The only method that is not improved by the data transformations is the Random Forest. The influence of K-means on the accuracy and prediction time for LinearSVC is shown in Figure 2.

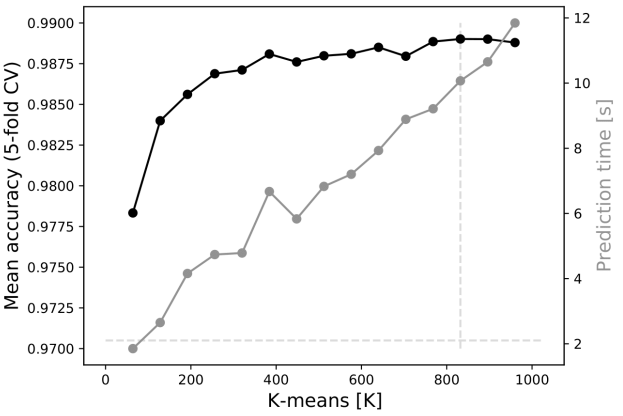

Figure 2: Mean accuracy and prediction time (5-fold CV) compared to different values of K for K-means for Kernel SVC. The optimal K for K-means is represented by the dashed vertical line ; the accuracy of the EukRep classifier by the horizontal one. Without K-means clustering, the accuracy is slightly smaller (0.9878) than the optimal K=832 (0.989) and the prediction takes sensibly more time (1.33-fold increase).

3.3 Fine-tuning of each method

Based on the results of the method fine-tuning, we find that the most accurate method is KernelSVC, with the 5-mer dataset transformed using K-means clustering of features, with $K = 832$, and parameters $C = 1000$ and $\gamma = 1500$, with 98.9% accuracy. The accuracy and prediction time of each method with the tuned parameters, as well as of the EukRep classifier, is shown in Figure 3.

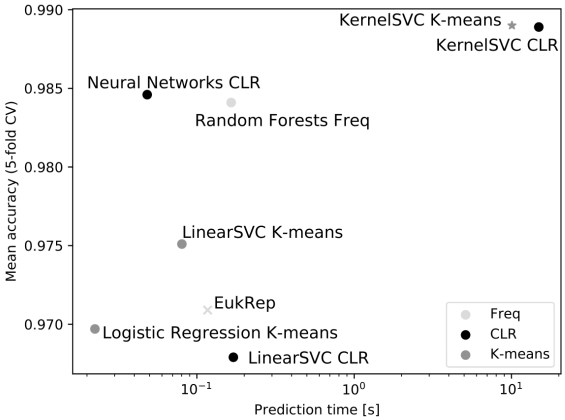

Figure 3: Mean accuracy compared to the prediction time (5-fold CV) on a log scale for the different methods with optimal hyper-parameters. The EukRep current method is shown by a cross marker. The best method in terms of accuracy is the Kernel SVC trained on K-means clustered data and is highlighted by a star marker. The shade represents the dataset the classifier was trained on: frequencies (light-grey), CLR transformed data (grey), K-means clustered features data (black).

4 Discussion

4.1 Influence of the k-mer size

As expected and reported in [11], k=5 represents a good trade-off between the accuracy of the classifier and the computing time. We consider that given the high accuracy of the methods we developed, the increase in computing time due to the increased number of features would not be worth the improvement in accuracy. However, for different problems using k-mer spectra of higher k values is useful, and we think that aggregating features using the K-means approach that we developed could vastly improve computing times.

4.2 Data Transformations

Here we show that for the Eukaryote/Prokaryote contig classification problem, data transformation through feature aggregation using K-means clustering or Centered Log Ratio standardisation vastly improves both the accuracy of the classifiers and computing times. For all methods except two, the data transformations that we implemented improve the accuracy. Interestingly, CLR transformation reduced the accuracy of the logistic regressions. Random Forests are not improved by CLR nor K-means aggregation of the features. For the CLR, this is expected as they are based on decision trees, their performance should not be affected by standardisation. For the K-means, we think that the improvement for most classifiers is linked to the aggregation of correlated features, allowing features uncorrelated to others to gain importance in the classification. Due to their nature, Random Forest are notably known to make use of all the feature space, concentrating decision trees for features that allow a good discrimination of the classes. This is why we don't see an improvement with the Random Forests using the K-means aggregation of features, the feature space is already well visited, and they do not need the help provided by this transformation.

Moreover, many methods showed reduced computing time when used with the transformed datasets. All methods show a decrease when using K-means clustered data, an improvement probably induced by the reduced number of features due to the clustering. For the CLR transform, the improvement can be explained by a faster convergence, helped by the log transformation. In some cases, this effect was of really high importance: the prediction time of Neural Networks is reduced by 9.84 fold when trained on the CLR data while improving the accuracy, as compared to a Neural Network trained on frequency data. If these findings are consistent and can be reproduced in other k-mer based classification problems (and they are numerous in computational biology), our approach could benefit many other methods. Particularly for problems that use larger k-mers sizes, that are now handled by removing k-mer with frequencies smaller than a threshold, as in [8]. The K-means feature aggregation, if allowed by a number of features not too large, could provide a better alternative to reduce the feature space.

4.3 Comparison of various machine learning classification algorithms

Our aim was to improve on the accuracy of the current LinearSVC. Many methods we developed had increased accuracy compared to the current method (see Figure 3). Kernel SVC classifiers performed with particularly high accuracies, the one trained on the K-means dataset being the best. This is consistent with literature, as many accuracy-oriented methods are currently based on Kernel-based Support Vector Machines, while speed-oriented ones rely on Linear SVM. In this study, we demonstrate that Neural Networks and Random Forests are good alternatives to the classical support vector methods. Per example, the Multi-Layer Perceptron used with CLR transformed data shows high accuracy and prediction times lower than the current method, making it potentially a good speed-oriented alternative. However, as for all Neural Networks based methods, the interpretation remains hard (see [3]) compared to the easier framework of Support Vector Machines. As in West et al. 2018, we report analogously better accuracies for the "Prokaryotes" class as compared to the "Eukaryote" one. This can potentially be explained by the higher "complexity" of the eukaryotic genomes (see [6]).

5 Conclusion

Here we present several classifiers that outperform current methods in terms of accuracy, computing time, or both. Notably, KernelSVC performed with really high accuracy (98.90% with 10.07 seconds prediction time) when trained on K-means clustered features. If a faster approach is needed, the Neural Networks trained on CLR transformed data performed with good accuracy (98.46%) while keeping the prediction time really low (0.0483 sec). We also demonstrate that K-means clustering for features aggregation and centered-log ratio standardisation of the data can highly improve both the accuracy of classifiers and the prediction time. These findings could have broad implications in k-mer based methods, widely used in many areas of bioinformatics, that are, to our knowledge, mostly based on k-mer frequencies without transformation. Thus, we plan to extend these findings to other DNA sequence classification problems. Finally, we will use and make available to other scientists in the metagenomics field the classifiers presented here that outperform state-of-the art methods on the NCBI genome dataset for the Eukaryote/Prokaryote contig classification problem.

Acknowledgement

This work is part of the "Vanishing Glaciers" project granted to TJB and supported by the NOMIS Foundation. We would like to thank Susheel Bhanu Busi for his precious help in building the training datasets and initial ideas. We extend our gratitude to Semih Gunel (the TA assigned to our group) that had precious Machine Learning insights. The experiments presented in this report were carried out using the HPC facilities of the University of Luxembourg.

Bibliography

[1] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting protein–protein interactions," *Bioinformatics*, vol. 21, no. suppl_1, pp. i38–i46, 2005.

[2] M. K. Faith, "Centered log-ratio (clr) transformation and robust principal component analysis of long-term ndvi data reveal vegetation activity linked to climate processes," *Climate*, vol. 3, no. 1, pp. 135–149, 2015.

[3] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3681–3688, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4252

[4] S. Khaleel, "Feature selection using k-means clustering for data mining," 03 2011.

[5] A. Li, J. Zhang, and Z. Zhou, "Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme," *BMC bioinformatics*, vol. 15, no. 1, p. 311, 2014.

[6] M. Lynch and J. S. Conery, "The origins of genome complexity," *Science*, vol. 302, no. 5649, pp. 1401–1404, 2003. [Online]. Available: https://science.sciencemag.org/content/302/5649/1401

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[8] N. T. Pierce, L. Irber, T. Reiter, P. Brooks, and C. T. Brown, "Large-scale sequence comparisons with sourmash," *F1000Research*, vol. 8, 2019.

[9] M. Tian, L. Hao, X. Zhao, J. Lu, and Y. Zhao, "The study of stream sediment geochemical data processing by using k-means algorithm and centered logratio transformation—an example of a district in hunan, china," *Geochemistry International*, vol. 56, no. 12, pp. 1233–1244, 2018.

[10] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, "Management of an academic hpc cluster: The ul experience," in *2014 International Conference on High Performance Computing Simulation (HPCS)*, 2014, pp. 959–967.

[11] P. T. West, A. J. Probst, I. V. Grigoriev, B. C. Thomas, and J. F. Banfield, "Genome-reconstruction for eukaryotes from complex natural microbial communities," *Genome research*, vol. 28, no. 4, pp. 569–580, 2018.