Towards a Systematic Investigation of Deep Learning Approaches for Bacterial Taxonomic Classification Using the 16S rRNA Gene

Yuju Ahn[2], Robbe Claeys[1], Moobeom Hong[1], Jihwan Lim[1] and Inkyun Park[1]
Supervisor: Joris Vankerschaver[3]

1. Universiteit Gent, Ghent 9000, BE
2. Imperial College London, SW7 2AZ London, UK
3. Ghent University Global Campus, Incheon 21985, ROK

Abstract. Modern bacterial taxonomy revolves around bioinformatics-based analysis, leading to deeper insights into microbial communities and their composition. The 16S ribosomal RNA (16S rRNA) gene is a frequently used and well-established phylogenetic marker for in silico bacterial classification. With the rise of sequence data, novel machine learning methods are required to deal with the increasing complexity involved in analyses. In this project, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) deep learning models were proposed to serve as efficient alternative approaches to bacterial classification. Machine learning models were trained and evaluated with a manually curated 16S dataset. Two sequence encoding strategies, k-mer and one-hot encoding, were studied and evaluated with the CNN- and RNN-based models respectively. Although a one-hot encoding approach allows for a greater variety of experimental comparisons, k-mer encoding showed superior results. The performance of deep learning models was compared against the conventional machine learning-based Ribosomal Database Project (RDP) Classifier in terms of accuracy and training time. The CNN model with 8-mer encoding showed 96.33% test accuracy at the genus level, 0.17%p higher than the RDP Classifier, showing the potential of deep learning approaches for bacterial classification.

Keywords: 16S rRNA, Bacterial Classification, Convolutional Neural Network, Machine Learning, Recurrent Neural Network.

1 Materials and Methods

1.1 Data Preparation

Three high-quality labeled 16S rRNA datasets, EzBioCloud [1], GRD [2] and 16S-UDb [3], were combined, curated and standardized. The resulting dataset contained 44,663 entries. To generate additional training data, an augmentation strategy was performed. Through addition of the biologically relevant reverse complement strands, the total number of entries in the training set was increased by 38,380.

The data was split into train, validation, and test sets in a stratified manner in a 6:2:2-ratio. Through stratification, the same number of unique labels in each set could be ensured.

Nucleotide sequences were encoded in two distinct ways: 1) a k-mer frequency encoding with sizes 3, 5, 7, and 8 and 2) a less computationally demanding one-hot encoding with three distinct maps, 1 regular (1, 0) map and 2 novel maps which considered mutation rates through penalty values [-1, 1].

1.2 Training

Four different deep learning models, with increasingly complex architectures that build on one another (CNN [4], BiLSTM, ConvBiLSTM [5] and an attention based ConvBiLSTM Read2Pheno [6]), were trained and tested on an AWS service with NVIDIA Tesla V100 GPUs.

Evaluations were set up to compare performance with various encoding strategies and between models at different taxonomic hierarchies.

2 Results and Discussion

The results showed that encoding played an important role in overall performance.

The CNN model scored highest on the longest k-mer tested the 8-mer. However, when comparing the 7- and 8-mer only a 1.09 %p increase in test accuracy with an almost 400 % increase in training time was observed.

All sequential models scored highest with the second mutation rate-adjusted one-hot encoding map, and the training time of the regular BiLSTM model was considerably longer than the other sequential models that performed an antecedent dimensionality reduction through convolution.

Overall, all models achieved test accuracies greater than 80% at the genus-level – a 954-class classification problem with a dataset of 44,663 entries (see Table. 1).

Deep learning-based classification models, in combination with informative and resourceful encoding methods, have demonstrated an outstanding capacity in learning the intrinsic features of biological data while maintaining flexibility, ease of use, speed and accuracy. Further experimentation based on the results of this study could facilitate the development of the ultimate bacterial taxonomic classifier, serving as a fundamental tool in the field of microbial research.

Table 1. Comparison of the evaluation metrics

Models & Encoding	Training time (m:s)	Test accuracy (%)	F1-score (%)	MCC
RDP 8-mer	00:31	96.16	96.22	0.9614
Conv BiLSTM one-hot map 2	05:31	82.88	81.97	0.8277
R2P one-hot map 2	17:20	88.92	88.16	0.8884
CNN 7-mer	01:51	95.24	94.64	0.9521

References

1. Yoon, S. H., Ha, S. M., Kwon, S., Lim, J., Kim, Y., Seo, H., & Chun, J.: Introducing EzBioCloud: a taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies. International journal of systematic and evolutionary microbiology, 67(5), 1613–1617 (2017).
2. Kim, S., Kim, S., Kenshiro, O., Wataru, S., Suguru, N., Todd D., T. & Masahira, H.: GRD: Curated Genomic-based 16S Ribosomal RNA Gene Database. *JDream III for advanced search and analysis*, *9*(89) (2015).
3. Agnihotry, S., Sarangi, A. N., & Aggarwal, R.: Construction & assessment of a unified curated reference database for improving the taxonomic classification of bacteria using 16S rRNA sequence data. *The Indian journal of medical research*, *151*(1), 93–103 (2020).
4. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings Of The IEEE*, *86*(11), 2278-2324 (1998).
5. Desai, H. P., Parameshwaran, A. P., Sunderraman, R., & Weeks, M.: Comparative Study Using Neural Networks for 16S Ribosomal Gene Classification. *Journal of Computational Biology*, *27*(2), 248–258 (2020).
6. Zhao, Z., Woloszynek, S., Agbavor, F., Mell, J. C., Sokhansanj, B. A., & Rosen, G. L.: Learning, visualizing and exploring 16S rRNA structure using an attention-based deep neural network. *PLOS Computational Biology*, *17*(9) (2021).

Full details on the implementation and results of this project can be found on the Jupyter notebooks of our GitHub repositories.

- Data: https://github.com/Lab-Vankerschaver/RNA-data
- ML models: https://github.com/Lab-Vankerschaver/16S-ML-models