


Review article

 Check for updates

Machine learning for microbiologists

In the format provided by the
authors and unedited

Supplementary Table 1. Software, packages, and libraries for ML in Microbiology. List of the main recommended stand-alone software, packages, and libraries with main characteristics, links, and most relevant features.

Type	Sub-type	Method	Package(s) and Link(s)
Supervised	Classification & regression	Support vector machine	- sklearn (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.svm) - LIBSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvm/)
Supervised	Classification & regression	Random forest	- sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html , https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)
Supervised	Classification	Bayes	- sklearn (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.naive_bayes)
Supervised	Classification & regression	k-Nearest neighbor	- sklearn (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.neighbors)
Supervised	Classification & regression	Adaptive boosting classifier	- sklearn (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble)
Supervised	Classification & regression	Deep learning	- TensorFlow (https://www.tensorflow.org/) - Keras (https://keras.io/) - PyTorch (https://pytorch.org/)
Supervised	Classification & regression	Neural network	- sklearn (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.neural_network)
Supervised	Regression	Logistic regression	- sklearn (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model)
Supervised	Regression	LASSO	- sklearn (https://scikit-learn.org/stable/modules/classes.html#regressors-with-variable-selection)
Supervised	Regression	ElasticNet	- sklearn (https://scikit-learn.org/stable/modules/classes.html#regressors-with-variable-selection)
Unsupervised	Dimensionality reduction	UMAP	- Python implementation (https://github.com/lmcinnes/umap)

			- R implementation (https://github.com/tkonopka/umap)
Unsupervised	Dimensionality reduction	t-SNE	- sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html) - R implementation (https://github.com/jkrijthe/Rtsne)
Unsupervised	Dimensionality reduction	PCA	- sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html) - R stats (https://www.rdocumentation.org/packages/stats/topics/prcomp) - FactoMineR (https://www.rdocumentation.org/packages/FactoMineR/)
Unsupervised	Dimensionality reduction	PCoA / MDS	- sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html)
Unsupervised	Dimensionality reduction	nMDS	- sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html)
Unsupervised	Clustering	Hierarchical	- sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html) - fastcluster (https://github.com/dmuellner/fastcluster/) - Scipy (https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html) - Cluster (https://www.rdocumentation.org/packages/cluster/)
Unsupervised	Clustering	Biclustering	- FABIA (https://www.bioconductor.org/packages/release/bioc/html/fabia.html) - biclust (https://www.rdocumentation.org/packages/biclust/) - QUBIC (https://www.bioconductor.org/packages/release/bioc/html/QUBIC.html) - sklearn (https://scikit-learn.org/stable/modules/biclustering.html)
Unsupervised	Clustering	Partitional	- K-means (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html) - K-medoids (https://github.com/kno10/python-kmedoids) - Scipy (https://docs.scipy.org/doc/scipy/reference/cluster.vq.html) - Cluster (https://www.rdocumentation.org/packages/cluster/)

Supplementary Box 1. Machine learning implementations

There are several machine learning implementations publicly available that are appropriate to be applied to microbiological data, starting from those that provide a graphical user interface, to command-line software and libraries to be used within a programming language. In general, the non-trained user might find more convenient the use of software with a graphical user interface, although, the more experienced and trained user could find more flexible and adaptable the direct use of software ML libraries via scripts and code. In the latter case, it is advisable to look for best coding practices and suggestions to make the writing, development, and debugging of the code easier¹⁻³. Graphical user interface tools are accessible to users without prior programming knowledge and are the easiest way for microbiologists to approach ML. Weka⁴ is a comprehensive software for ML accessible via a graphical user interface as well as via application programming interfaces (APIs) and R/Python wrappers. MicrobiomeAnalyst^{5,6} is a web-based framework that allows for multi-level analysis of microbiome data, among which there is the classification using Random Forest. QIIME 2⁷ and mothur⁸ are free and open-source platforms specifically for microbiome analysis with ML capabilities that can be extended with plugins developed by the community. Within the QIIME 2 platform, there are two plugins that implement machine learning frameworks “q2-feature-classifier”⁹ which implements machine learning methods for sequence classification, and “q2-sample-classifier”¹⁰ which implements supervised machine learning methods to predict sample outcomes using microbiome composition or other quantitative data as input features. Finally, SIAMCAT is a statistical inference and machine learning pipeline implemented in R for the identification of biomarkers.¹¹

Command line software and libraries generally require users to have some programming capabilities and will be less approachable to novice users than tools with a graphic interface. However, software and libraries are more widely applicable to many different scenarios. For the Python programming language, the most popular machine learning library is scikit-learn¹², an open-source package providing both supervised and unsupervised approaches and many functionalities for data preprocessing and feature selection. Packages focused on deep learning include TensorFlow¹³, Keras, and PyTorch¹⁴. Other implementations for dimensionality reduction and clustering can be found in the SciPy library¹⁵ and specific packages, i.e. UMAP (<https://github.com/lmcinnes/umap>). The R programming language has the mlr3 and tidymodels packages^{16,17}, but many other machine learning approaches are available in specific packages such as “caret”, “stats”, “cluster”, “FABIA”, and “QUBIC”. Additionally, some of these libraries have been enriched with AutoML packages that automate the process of model selection and hyperparameter tuning and find the most appropriate configuration for the predictive task at hand. Popular examples include AutoSklearn¹⁸ and AutoKeras¹⁹ for scikit-learn and Keras, respectively.

Supplementary Box 2. Causal inference versus prediction

Causal inference is a frequent objective of observational studies for which euphemisms are not helpful²⁰. ML studies that are mainly concerned with developing accurate prediction models do not depend on knowing causal relationships to meet this objective; however, complex or unknown causal relationships can impact generalizability and potential for the success of microbiome-based interventions. Specifically for the case of ML applied to human microbiome data, some potential causal relationships²¹ consistent with accurate prediction models include:

- microbiome -> outcome: the most desirable relationship for generalizability and for the potential of interventions on the microbiome to alter the outcome. However, even in the

presence of such a causal relationship, generalizability is not assured if there are mediators of the effect of the microbiome on the outcome, or other factors also affecting the outcome, that differ between the sample and target population.

- outcome → microbiome: considerations for reverse causality in prediction are similar, but interventions on the microbiome would not alter the outcome. Mediators may still be present, such as treatments of the outcome that affect the microbiome
- microbiome ← confounder → outcome: confounding occurs when a common factor causes both the outcome and the microbiome predictor, inducing correlation. If the confounder is known and observed, controlling for it in a regression model can estimate the association between predictor and outcome in the absence of confounding. Unmeasured confounding is difficult to account for, but a validation of models in different populations where the prevalence or effect of unmeasured confounders is different may give an idea of the impact of confounding.
- microbiome → collider ← outcome (direct arrow between microbiome and outcome not shown): A collider is caused by both the microbiome and the outcome of interest. Colliders are not necessarily a problem but can present a methodological pitfall because controlling for a collider will *introduce* bias in the estimated causal effect between microbiome predictor and outcome.

Standard epidemiological methods for causal inference²², such as adjustment in a multivariate regression model, can be applied to the predictions made by machine learning as a single variable instead of the hundreds of variables present in a typical microbiome dataset. Causal inference remains challenging and usually requires experimental validation, but these methods provide a framework for dealing with the challenges in observational datasets.

References

1. Djaffardjy, M. *et al.* Developing and reusing bioinformatics data analysis pipelines using scientific workflow systems. *Comput. Struct. Biotechnol. J.* **21**, 2075–2085 (2023).
2. Rule, A. *et al.* Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLoS Comput. Biol.* **15**, e1007007 (2019).
3. Topçuoğlu, B. D., Lesniak, N. A., Ruffin, M. T., 4th, Wiens, J. & Schloss, P. D. A Framework for Effective Application of Machine Learning to Microbiome-Based Classification Problems. *MBio* **11**, (2020).
4. Hall, M. *et al.* The WEKA data mining software. *ACM SIGKDD Explorations Newsletter* vol. 11 10–18 Preprint at <https://doi.org/10.1145/1656274.1656278> (2009).
5. Dhariwal, A. *et al.* MicrobiomeAnalyst: a web-based tool for comprehensive statistical, visual and meta-analysis of microbiome data. *Nucleic Acids Res.* **45**, W180–W188 (2017).
6. Chong, J., Liu, P., Zhou, G. & Xia, J. Using MicrobiomeAnalyst for comprehensive statistical, functional, and meta-analysis of microbiome data. *Nat. Protoc.* **15**, 799–821 (2020).
7. Bolyen, E. *et al.* Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nat. Biotechnol.* **37**, 852–857 (2019).
8. Schloss, P. D. Reintroducing mothur: 10 Years Later. *Appl. Environ. Microbiol.* **86**, (2020).
9. Bokulich, N. A. *et al.* Optimizing taxonomic classification of marker-gene amplicon sequences with QIIME 2's q2-feature-classifier plugin. *Microbiome* **6**, 90 (2018).
10. Bokulich, N. A. *et al.* q2-sample-classifier: machine-learning tools for microbiome classification and regression. *J Open Res Softw* **3**, (2018).
11. Wirbel, J. *et al.* Microbiome meta-analysis and cross-disease comparison enabled by the

- SIAMCAT machine learning toolbox. *Genome Biol.* **22**, 93 (2021).
12. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
 13. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv [cs.DC]* (2016).
 14. Paszke, Gross, Massa & Lerer. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.*
 15. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
 16. Lang, M. *et al.* mlr3: A modern object-oriented machine learning framework in R. *J. Open Source Softw.* **4**, 1903 (2019).
 17. Kuhn, M. & Wickham, H. Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. *Boston, MA, USA*.
 18. Feurer, M. *et al.* Auto-sklearn: Efficient and Robust Automated Machine Learning. *Automated Machine Learning* 113–134 Preprint at https://doi.org/10.1007/978-3-030-05318-5_6 (2019).
 19. Jin, H., Song, Q. & Hu, X. Auto-Keras: An Efficient Neural Architecture Search System. in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 1946–1956 (Association for Computing Machinery, 2019).
 20. Hernán, M. A. The C-Word: Scientific Euphemisms Do Not Improve Causal Inference From Observational Data. *Am. J. Public Health* **108**, 616–619 (2018).
 21. Pearl, J. *Causality: Models, Reasoning, and Inference*. (Cambridge University Press, 2000).
 22. Pearl, J. Causal inference in statistics: An overview. *Stat. Surv.* **3**, 96–146 (2009).