ADVANCING STANDARDS FOR MACHINE LEARNING IN THE BIOSCIENCES

by

Kenneth E. Schackart III

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF BIOSYSTEMS ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2022

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by: Kenneth E. Schackart III, titled:

Advancing Standards for Machine Learning in the Biosciences

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Bonnie Hurwitz — Date: Nov 22, 2022

Timothy Secomb — Date: Nov 23, 2022

Lingling An — Date: Nov 23, 2022

Jana U'Ren — Date: Nov 28, 2022

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Bonnie Hurwitz — Date: Nov 28, 2022
Dissertation Committee Chair
Department of Biosystems Engineering

ACKNOWLEDGEMENTS

I would like to thank my advisor, committee members, and mentors. Thank you to my advisor, Dr. Hurwitz, for your support and guidance. Thank you to my committee members, Dr. Secomb, Dr. An, and Dr. U'Ren, for challenging and guiding me during my Ph.D. Thank you Dr. Yoon for your mentorship and guidance during my undergraduate and graduate studies. Thank you Dr. Ponsero for your incredible mentorship, support, and feedback during my Ph.D. Thank you Ken Youens-Clark for your mentorship and inspiration. Thank you, Dr. Heidi Imker, for your support and guidance. Thank you to all the others in my life who have helped guide and advise me during my studies.

I would like to my supporters and collaborators. Thank you, Dr. Chuck Cook at the Global Biodata Coalition, for your support, encouragement, and opportunities during my Ph.D. Thank you to my collaborators at the Chan Zuckerberg Initiative, especially Ana-Maria Istrate, Dr. Gully Burns, and Dr. Donghui Li for your incredible collaboration and all I have learned from you. Thank you to the many people and groups that have supported my studies.

I would like to thank my colleagues and friends. Thank you to all current and former members of Hurwitz Lab and Yoon Lab for your feedback, friendship, and exciting conversations, especially Dr. Kattika Kaarj, Dr. Alex Day, Babak Safavinia, Dr. Sangsik Kim, Dr. Tiffany Ulep, Dr. Lane Breshears, Patarajarin Akarapipad, Katherine Sosnowski, Yan Liang, Dr. Kai Blumberg, Dr. Chunan Liu, and Matthew Miller. Thank you, Jilian, Garrett, Mara, and Max, for your friendship and care.

Importantly, I would like to thank my family. Thank you, Mom, for your unceasing support, encouragement, and guidance. You have made everything possible and made me feel that I could do anything. Thank you to my siblings, Bailee, Danielle, and Kalub, for your

encouragement and for helping shape who I am today. You have given me the strength to tackle any challenge I face. Thank you, David, for your support and encouragement. Thank you, Tid, for your support, love, and encouragement. You inspire me to be the best I can be. Thank you to all my incredible family for your love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The data revolution in the life sciences has brought on new challenges and opportunities. Large-scale and complex data from novel sensing mechanisms challenges existing infrastructure and methods of modeling and drawing inferences. Increasingly, researchers turn to machine learning (ML) methods to detect patterns and draw inferences from these datasets. While these methods have brought new insights and capabilities to process these data, new pitfalls have emerged. In many cases, the reproducibility of these studies is challenging as standards continue to evolve. Further, ML methods are being developed at a rapid rate, making it difficult for researchers to assess the utility and limitations of new methods. New standards are needed to improve the reproducibility, reusability, and comparison of ML and other computational methods. The aim of this dissertation is to advance those standards and drive the comparison of tools and analytics. Specifically, three works are presented, each of which aims to further reproducibility standards regarding the use of ML in life science research.

The first chapter is a review paper that explores the use of ML for biosensing using bioreceptor-free biosensors. In traditional biosensors, the bioreceptor provides specificity and sensitivity to the sensor. The bioreceptor, being of biological origin (*e.g.,* an enzyme) degrades over time and under certain ambient conditions. To circumvent this, sensors have been developed which attempt to imitate the bioreceptor and biological sensing organs. In many cases, the loss of performance due to the lack of bioreceptor is compensated with robust data analysis techniques, particularly ML for pattern detection. Given the rapid pace of ML development, the methods used for each sensing modality varies, as does the performance of those methods. Additionally, the methods used for different sensing modalities are disjoint, even when similar data structures and problems are faced in various subfields. In this review, the methods that have been used for the

different sensing modalities, and the ones that have been the most performant are highlighted. By addressing the field as a whole, this review provides guidance to researchers regarding comparisons between candidate models and points them toward future directions.

The second chapter goes beyond a comparison of reported performance and conducts a thorough benchmarking of computational tools developed for detecting bacteriophage (phage) sequences in metagenomes. Sequence classification, particularly for detecting phage, is a difficult task. Phages rapidly mutate, lack a conserved biomarker, and can integrate into the host genome as a prophage. By July of 2021, at least 19 computational tools have been published for detecting phage sequences in metagenomes, each taking a different approach to the problem. Many of the tools build ML models using features related to homology (*e.g.,* viral hallmark genes, lack of bacterial genes) or features of the sequences themselves (*e.g., k*-mers). When attempting to run these tools, less than half (9 / 19) could be run at scale, if at all. Of the tools compared, performance on several metrics varied significantly. The strengths and weaknesses of the tools are evaluated to inform future research. All datasets used for benchmarking the tools are made available to facilitate further benchmarking efforts.

Finally, the last chapter is an opinion paper presenting the methods used and rationale for implementing a ML-enabled workflow that reaches a high level of reproducibility. In this work, considerations for reproducibility, code, and data standards are discussed. The paper describes how standards were implemented, including which tools and software packages were used. This work serves as an example for implementing a reproducible ML research project and opens further discussion on how the implementation could be further improved.

1. INTRODUCTION

Most scientific fields, and in particular biosciences, are currently facing a data revolution, and with it comes unique challenges and opportunities. One example of this is the field of genomics and metagenomics. In 2006, the cost to generate a high-quality draft human genome was approximately 14 million dollars while in 2021 the cost has dropped to about $1,000 (Wetterstrand, 2021). Recent advances in high-throughput sequencing technologies have accelerated genomic and metagenomic research, but the volume of data generated, and their complexity creates their own set of challenges. Similarly, in very different fields like biosensor research, sensor data has drastically evolved from classical biosensors such as the glucose-meter and pregnancy test to novel bioreceptor-free methods that often rely on non-specific electrochemical arrays or imaging (Ozer & Henry, 2021; Rivenson, Liu, et al., 2019; Tan & Xu, 2020). Both of these novel sensing modalities make traditional methods of interpretation, such as regression and standard curves, ineffective. This creates novel challenges in analyzing and interpreting the data from these sensors.

Advanced statistical and machine learning (ML) methodologies are designed to meet some of these challenges and have been more and more prevalent in data analysis of biological datasets. These methods are indeed particularly relevant in this context of large and complex data. As a result, biological sciences have witnessed a rise of ML methods to work with these complex datasets.

1.1. Machine Learning Applications in Biological Sciences

Machine learning (ML) is a terminology associated with a varied number of computer systems that are able to learn and adapt without following explicit instructions. This is particularly useful when datasets are large, and the patterns in them are complex or unknown. In life sciences these methods are employed extensively to detect patterns in the data and make predictions based

on those patterns (Koohy, 2018). Before discussing examples of ML applications in biological sciences, it is necessary to introduce some key concepts.

1.1.1. The Seven Steps of Machine Learning Methods

1.1.1.1. Data Collection

The first step of an ML project is to collect data relevant to the desired application. This can include taking measurements of physical properties or assigning labels to observations (*e.g.,* labeling emails as spam vs not spam) (Paleyes et al., 2022). It is important during data collection to minimize noise and error, and to collect data that is representative of the intended application. For instance, if the task is to classify news articles by category, there must be sufficient examples from all anticipated categories (Hasib et al., 2020).

The potential performance of the final model is dependent on the quality and quantity of data since this is the sole source of information from which the model can learn. Since data collection can be costly and time consuming, it is often the limiting factor of the final model's performance.

1.1.1.2. Data Preparation

Once the data are collected, they must be curated and prepared for analysis. During data preparation, the quality of the data can be assessed. For instance, if there is a category imbalance (more observations of certain classes than others), subsampling may be used to restore balance (Hasib et al., 2020). Exploratory data analysis may also be used to detect anomalies and check assumptions to guide model selection. For example, some models are designed with the assumption that each variable has values that are normally distributed, and if they are not, it may be necessary to transform the data (*e.g.,* taking the logarithm) such that they meet the assumptions (Osborne, 2019).

Additionally, each model requires the input variables, referred to as features, to be in a particular form. The process of obtaining those features from the raw data is called feature extraction. Since ML models are fundamentally mathematical models, the features generally must be converted to numerical representations of the data. For instance, text data like an email may be converted to a vector, where each position represents a word in the vocabulary, a value of 1 indicates that word is present in the email, and a value of 0 indicates that word is not present (Brownlee, 2017). This so-called one-hot encoding of the text is the feature vector which is used for learning (Arnaud et al., 2021).

Finally, the data are split into several partitions. The simplest approach is splitting between data used for training the model (training set), and data used for evaluating the model (test set). It is common to split with 70-80% assigned to training, and 20-30% assigned to test (Joseph, 2022). The test set should be withheld during model training to avoid biasing the training process. Prior to training, it is common to again split the data into training and validation splits. During each training step, parameters of the model are adjusted using the training split, and the quality of the prediction is evaluated on the validation set. The performance on the validation set is used to further adjust model parameters in the next step of training. Splitting between training and validation can be done similarly to the initial training-validation split, or more robust approaches can be used such as *k*-fold cross validation (CV) (Picard & Cook, 1984). In *k*-fold CV, the training data are split into *k* groups, and during each training step all but one group are used for training, and the last group is used as the validation group. This process repeats, where the validation group changes each time, such that at the end, all sets have served as the test set (Hastie et al., 2001).

1.1.1.3. Model Selection

Model selection is performed based on the application, data type, and constraints. Types of models are further discussed in section 1.1.2. The model selection should be driven by the intended model application, such as classification task, anomaly detection or class prediction (Paleyes et al., 2022). Importantly, the model must also be appropriate for the data type. Indeed, models are designed for certain types of data, and may make assumptions about the data. For example, linear discriminant analysis assumes that input variables are numeric and homoscedastic (have the same variance) (Izenman, 2008). Finally, an additional consideration is selecting a model that can operate within imposed constraints (Paleyes et al., 2022). For example, some models require specific types of computing hardware, like a graphics processing unit (GPU) for training. Depending on the application of the ML system, speed and computing efficiency may be critical as in real-time applications like robotics or autonomously driving a vehicle (Paleyes et al., 2022).

It is common to select several candidate models, and evaluate them alongside one another before making a final selection (Maeda, 2018).

1.1.1.4. Training

Model training is where the "learning" occurs. Model parameters are initialized to some value, which could be random numbers, or constant values. Those parameters are iteratively changed during training steps (Schmidhuber, 2015). During each training step, the training set is provided to the model, the model performs the calculations based on the model architecture and parameters, and the output is created. The output is compared to a desired output, and the differences between the desired and actual outputs are quantified (*e.g.,* using a loss function). Model parameters are then updated, and another training step begins. Training may be run for a predetermined number of steps, or until a certain level of prediction performance is achieved.

1.1.1.5. Evaluation

Once training is complete, the model's performance is evaluated on data not used for training, such as the test set. To do so, the test set is passed to the model and predictions are made (Paleyes et al., 2022). The predictions are often evaluated using performance metrics (Agrawal, 2021), such as those presented in section 1.1.2. If multiple models are being compared, they are evaluated against each other. At this stage, it is determined whether the model performance is sufficient for the application. If it is not, more data may need to be collected, data preparation may need to be improved, or another model may be selected.

1.1.1.6. Hyperparameter Tuning

If a model performs well during evaluation, steps are taken to continue to improve it. One way of doing this is modifying how the parameters are initialized. Parameter initialization can affect model performance (Paleyes et al., 2022). For instance, if parameters are tuned by minimizing some function, then the ideal parameter would lead to the global minimum of that function. However, that function may also have other local minima. Depending on the initial parameter value, the training may only find a local minimum of the function and not the global minimum as desired.

1.1.1.7. Real-world Deployment

Finally, the model can be deployed and applied to real world datasets. When using a supervised predictive model, performance is assessed on the withheld test set to estimate how it will perform for its intended use-case (Paleyes et al., 2022). It is often necessary to periodically evaluate model performance in deployment. A threat to ongoing model performance is concept drift, where data distributions shift over time, making the training sets less representative of incoming data (Leon-Medina et al., 2020).

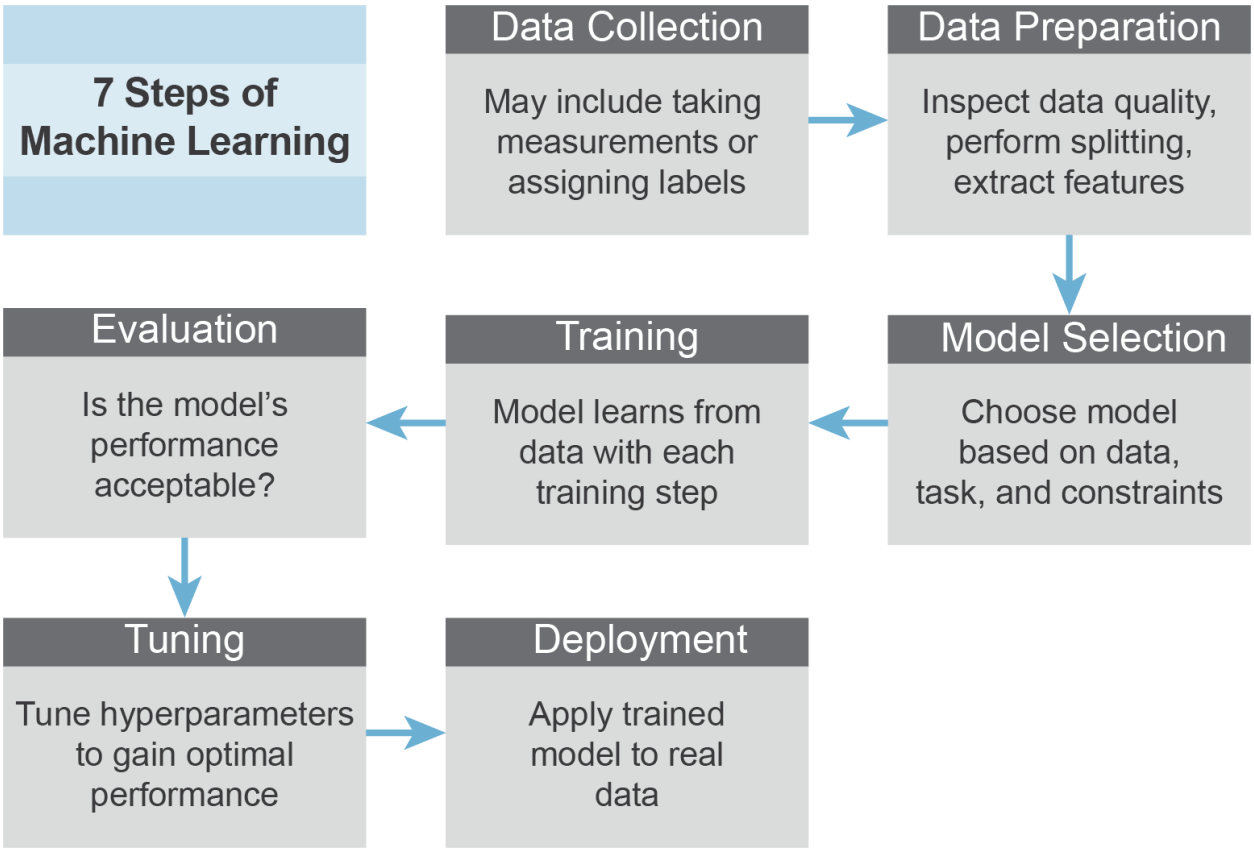

Figure 1

Overview of the steps of machine learning as a flowchart from data collection to deployment.

1.1.2. Machine Learning Key Concepts and Models

1.1.2.1. Supervised vs. Unsupervised Methods

Most ML methods can be broadly categorized as either supervised or unsupervised. In supervised models, the goal is typically to make a prediction of a ground-truth property often determined experimentally (Rajoub, 2020). Training data is annotated with labels, most commonly class (no inherent order), ordinal (ordered class labels), or numerical labels. During the training of supervised models, the model is optimized to predict the label by identifying patterns in the data.

On the other hand, the goal of unsupervised methods is to recognize patterns in data without labels. These methods aim to recognize similarities in complex datasets without the need of a ground-truth label.

Finally, semi-supervised methods will merge these two approaches. During training, a small number of labeled observations are combined with a larger number of unlabeled observations. This approach can help offset the costs of obtaining a sufficient amount of labeled data (Zhu & Goldberg, 2009).

1.1.2.2. Dimensionality Reduction

Dimensionality reduction methods are a group of unsupervised machine learning models that aim to reduce a large number of variables (or features) to a smaller number of features while minimizing information loss. Ideally, observations that were similar in the original feature space will still be similar in the reduced feature space (*e.g.,* protein expression of closely related organisms), while dissimilar observations will remain dissimilar (*e.g.,* protein expressions of distant organisms) (Cunningham, 2008). Some of the most common dimensionality reduction algorithms are principal component analysis (PCA), linear discriminant analysis (LDA), and *t*-distributed stochastic neighbor embedding (*t*-SNE) (Cunningham, 2008).

Dimensionality reduction is useful when dealing with high-dimensionality data composed of many correlated features. Correlated features are redundant, meaning that they contain similar information, and bloat the feature space (Cunningham, 2008). For example, if the task is to predict the weight of a strawberry based on an image, seven features may be the color (in terms of red, green, and blue saturation values), the number of seeds, the width, the length, and the area of the strawberry in an image. Many or all these features may be correlated, where heavier fruit tend to be redder, have more seeds, and be larger in size. PCA could likely reduce those seven features to

two that can be plotted on two axes and still capture most of the information contained in the original features.

1.1.2.3. Clustering, Classification, and Regression Tasks

Clustering algorithms are a family of unsupervised ML methods that group data based on similarities in their features. For example, k-means is a classic powerful clustering algorithm, and is used for a variety of data types (Jain, 2010). k-means groups data into k clusters, where k is a hyperparameter. As with all hyperparameters, the choice of k can affect the quality of results. k-means is a centroid-based clustering algorithm, other clustering algorithms are density based (clustering areas of high density), distribution based (assuming observations of each group follow a distribution such as Gaussian distribution) or hierarchical (grouping at several hierarchical levels). Hierarchical clustering is particularly useful when it is known that data are hierarchical in nature, such as clustering genomic data and the predicted clusters may correspond to taxonomic relationships. When little is known about the observations prior to analysis, clustering can help identify patterns for further analysis.

On the other hand, classification is a supervised ML task that outputs predictions in the form of class labels (*e.g.,* disease status from sample characteristics). Although probabilistic classifiers predict class labels based on a threshold from a numerical output, they are trained on categorical labels, and prediction scores correspond to those labels. There are many popular classification algorithms, such as k-nearest neighbors (k-NN), support vector machines (SVM), and decision trees (DT) and random forest (RF) (Martens et al., 2011).

Classification performance can be measured using many performance metrics. For binary classification, conventionally one class is considered ‘positive’ while the other is considered ‘negative’. For instance, if the task is to detect spam emails, a spam email would be positive, while

a non-spam email would be negative. Consequently, a spam email correctly predicted to be spam is a True Positive (TP), while a non-spam email sequence correctly labeled as such is a True Negative (TN). Conversely, a non-spam email incorrectly classified as spam is a False Positive (FP), and a spam email incorrectly classified as non-spam is a False Negative (FN). Sensitivity, also known as recall, is the proportion of all positives that were correctly identified (eq. 1). Precision is the proportion of all predicted positives that are indeed positives (eq. 2). Specificity is the proportion of all negatives that are correctly labeled as negatives (eq. 3). Accuracy is the proportion of correct labels out of all predictions (eq. 4) and is a poor metric on imbalanced data (data where the numbers of positives and negatives are different). *F*1 score is the harmonic mean of precision and sensitivity (eq. 5) and is a better metric for imbalanced data (Agrawal, 2021).

$$sensitivity\ (recall) = \frac{TP}{TP + FN} \tag{1}$$

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$specificity = \frac{TN}{TN + FP} \tag{3}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$F1 = \frac{2 \times precision \times sensitivity}{precision + sensitivity} \tag{5}$$

Finally, regression models are designed to output numeric predictions. Generally, this value is continuous (*e.g.,* glucose concentration in blood), but may be ordinal, as in the case of ordinal regression (*e.g.,* pain on a scale from 1-10) (Winship & Mare, 1984). Traditional regression models include multilinear regression (MLR), support vector regression (SVR) which is an adaptation of SVM, and regression tree which is an adaptation of decision tree.

Regression models are trained by minimizing a type of error, such as the sum of squared residuals. There are a variety of metrics for assessing regression performance, such as root mean

squared error (RMSE), mean absolute error (MAE), normalized root mean squared error (NRMSE), and coefficient of determination R^2 (Botchkarev, 2019).

1.1.2.4. Artificial Neural Networks

Recently, a family of machine learning algorithms have been developed based on the artificial neural network (ANN), named for its inspiration from biological neural networks. These models can be used for a large variety of tasks, including classification and regression, and beyond into complicated tasks such as protein conformation prediction (Jumper et al., 2021). The classic ANN is a multilayer perceptron, in which layers of artificial neurons are interconnected. Individual neurons can be "activated" based on their inputs such as input features or outputs of other neurons. This basic layout has been extended to include structures such as back propagation, convolution, recurrence, and attention. Recently, the latter case has led to powerful models known as transformers which have shown state-of-the-art performance on a variety of tasks (Vaswani et al., 2017) such as protein structure modeling (Jumper et al., 2021).

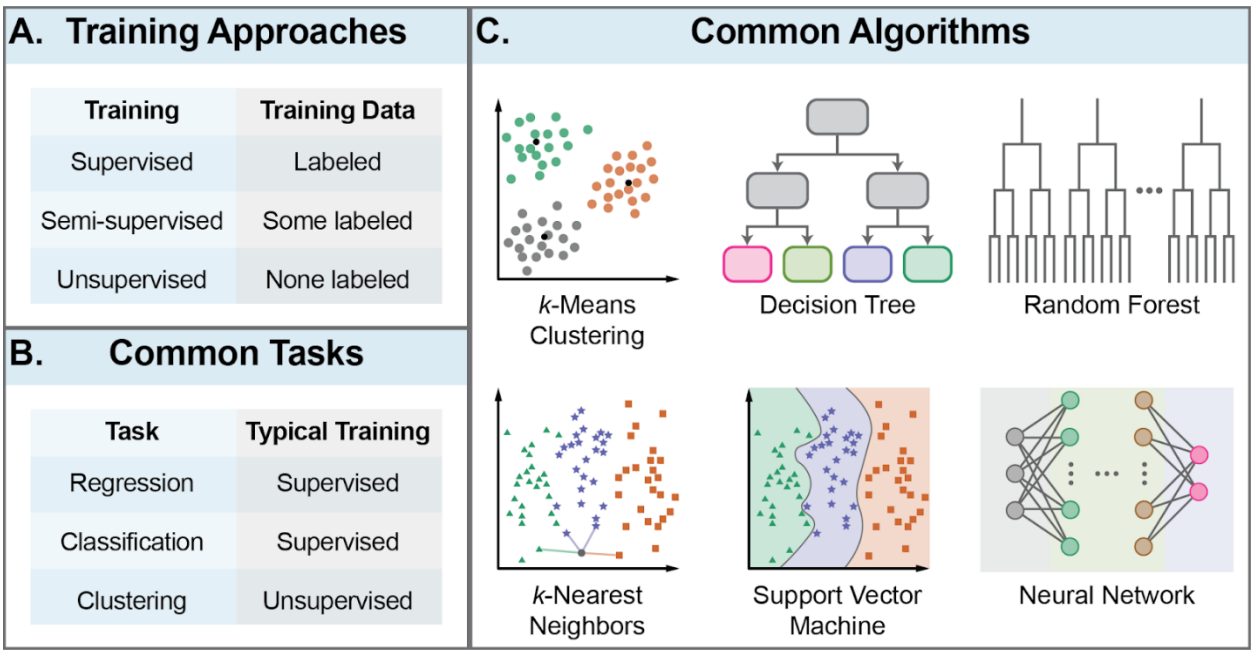

Figure 2

Overview of ML concepts, including training approaches, common tasks for ML, and common algorithms that are used. A) Table defining supervised, semi-supervised, and unsupervised training approaches regarding labeled training data. B) Common tasks of machine learning and how they are typically trained. C) Graphical representations of six common ML algorithms. Portions of panel C are modified from Schackart and Yoon 2021 under Creative Commons Attribution License.

1.1.3. Examples of Machine Learning Applications in Biosciences

From simple dimensionality reduction to transformers, ML models can identify patterns in complex and large-scale datasets. Consequently, these models are particularly promising to analyze complex and large-scale biological datasets. Here we review in detail two ML applications of interest for this dissertation: The case of Bioreceptor-free biosensors and sequence classification from metagenomic datasets.

1.1.3.1 Bioreceptor-Free Biosensors

Biosensors are sensors which use a biological receptor (bioreceptor) element to facilitate sensing of target analytes. Bioreceptors are selected for their inherent specificity to target analytes. Common bioreceptors are enzymes (H. H. Nguyen et al., 2019), antibodies (Hock, 1997), and aptamers (Lim et al., 2010). The interaction between the bioreceptor and target analyte ultimately leads to the generation of a signal which is detected by a transducer. An example of this can be seen with the glucose meter. Glucose oxidase, an enzyme, is the bioreceptor, which binds to and oxidizes glucose. A series of reduction-oxidation (redox) reactions occur which generate an electrical current at the electrode that serves as the transducer (Yoon, 2016).

Biosensors can be broadly categorized based on the sensor signal type. Electrochemical biosensors, like the glucose meter, rely on a chemical reaction to produce an electrical signal, typically via redox reaction, which is detected by the transducer. Optical biosensors generate an electromagnetic signal, such as a color indicator (Edachana et al., 2019; Y. Mutlu et al., 2017), fluorescence (Lin et al., 2018; Song et al., 2015), luminescence (Chen et al., 2018), surface plasmon resonance (X. Guo, 2012), and light scattering (Heinze & Yoon, 2011).

Biosensors have been applied to detecting and quantifying a wide variety of analytes such as human chorionic gonadotropin as a pregnancy test (Wide & Gemzell, 1960), disease biomarkers (Mahmudunnabi et al., 2020), bacteria (Ahmed et al., 2014), and environmental pollutants (Gavrilescu et al., 2015). Many of these applications are of clinical and economic importance.

While the bioreceptor provides specificity and sensitivity to the biosensor, it also has several limitations. Since bioreceptors are biological elements, they are often prone to degradation over time and under certain conditions (Massah & Asefpour Vakilian, 2019). This can lead to drift of the sensor signal yielding inaccurate measurements, or reduced sensitivity. To address these challenges, there has been much attention paid to developing sensors that are inspired by bioreceptors and biological sensing organs (*e.g.,* the tongue), while eliminating the bioreceptor. Without a bioreceptor, a new set of challenges emerges, especially regarding sensor specificity. With less specific sensor signal generation, researchers turn to ML to detect patterns in the sensor data that are specific to the analyte.

1.1.3.1.1. Electrochemical Bioreceptor-Free Biosensors

Bioreceptor-free electrochemical biosensors frequently derive inspiration from biological sensing organs such as the animal nose and tongue. A tongue has taste receptors that detect, with low specificity, certain chemicals (Bachmanov et al., 2014). The signals from these various

receptors are processed by the brain to identify the identity of food and beverages. The Electronic tongue (Etongue) is modeled after this scheme, where an array of electrochemical sensors with low specificity generates a multivariate spectral signal in response to interaction with chemicals in solution. Since the signal is non-specific, ML is used to detect the patterns that are indicative of analyte presence and concentration, or more general attributes like solution classification (*e.g.,* tea vs. coffee). Similar electrochemical arrays are used for detecting volatile compounds in the air and are referred to as electronic noses (Enose).

Since the electrochemical sensors are non-specific, there tends to be redundancy in the data (*i.e.,* multiple sensors in the array responding to the same chemical). Therefore, PCA is frequently used to remove redundancy and reduce the dimensionality of the data. A common task of Enose and Etongue is identifying scent or taste, respectively, which are classification problems. SVMs have been used extensively to classify the dimensionally reduced array data (Chen et al., 2018; Tan & Xu, 2020). Other methods, particularly ANNs, work well on the full sensor array dataset (Hotel et al., 2018; Sanaeifar et al., 2017). Additionally, transfer learning (updating a trained model's parameters to account for changes in data distributions) has been extensively employed to counter Enose sensor drift, reducing the need for complete retraining (Leon-Medina et al., 2020; B. Liu et al., 2019; Yi et al., 2021).

1.1.3.1.2. Optical Bioreceptor-Free Biosensors

Most bioreceptor-free optical biosensors utilize spectroscopy or imaging. Of the spectroscopic biosensors, surface-enhanced Raman spectroscopy (SERS) has shown great success (T. Yang et al., 2015; Zhang et al., 2018). SERS is a vibrational surface sensing technique that enhances Raman scattering based on surface characteristics. Briefly, SERS utilizes incident laser light to induce inelastic scattering (Raman scattering) from the target analyte. The intensity of the

Raman scattering is enhanced by interaction with the conduction electrons of metal nanostructures on the surface of the substrate (SERS substrate). The enhancement of the Raman scattering is what makes SERS so sensitive. While SERS can generate signals in response to very low analyte concentrations, the signal can be difficult to interpret. One reason is that the scattering properties of a specific analyte are also dependent on the surface characteristics. Therefore, there are no universal methods of interpreting SERS spectra.

Like the signals from sensor arrays in Enose and Etongue, there is some level of redundancy in SERS spectra. Scattering of similar wavelengths is often similar. To reduce this redundancy, PCA is often used as a dimensionality reduction step (Hassoun et al., 2018; Hidi et al., 2016a; Tan & Xu, 2020). Classification of the target analyte is often done using SVM on the PCA transformed data (Hassoun et al., 2018; Sun et al., 2019).

Another sensing technique among optical biosensors is imaging. Imaging utilizes optical sensor arrays, such as a complementary metal oxide semiconductor (CMOS) sensor for data collection. In order to develop compact and low-cost imaging-based bioreceptor-free biosensors, researchers have developed lens-free sensors (Roy et al., 2017). Without a lens, the images are out of focus, and must be reconstructed computationally. Convolutional neural networks (CNN) have been used extensively for lens-free imaging reconstruction (Rivenson, Wu, et al., 2019). Further, CNNs have been used to detect signs of analyte presence in these devices, such as microparticle aggregation caused by interaction with viral particles from herpes simplex virus (Wu et al., 2019).

Many other applications use other imaging techniques such as quantitative phase imaging (Rivenson, Liu, et al., 2019) and Fourier transformed infrared spectroscopy (FTIR) (Lotfollahi et al., 2019). One particular area of note is in digital staining. Microscopic phase or FTIR images are taken of histological samples, without immunostaining, and the structures in the tissue are

differentially colorized computationally. To do so, CNN models are trained on immunostained samples (*e.g.,* hematoxylin and eosin stain), to learn the patterns indicative of how the tissues become stained. The model can then make predictions of how an unstained section would look if it were stained.

Additionally, due to the labor and time intensive cost of collecting imaging training data, generative adversarial networks (GAN) have been used to bolster the training process. GANs create artificial images based on real images in the training set, effectively increasing the training set size, which can improve model performance and help overcome the challenges to obtaining labeled data.

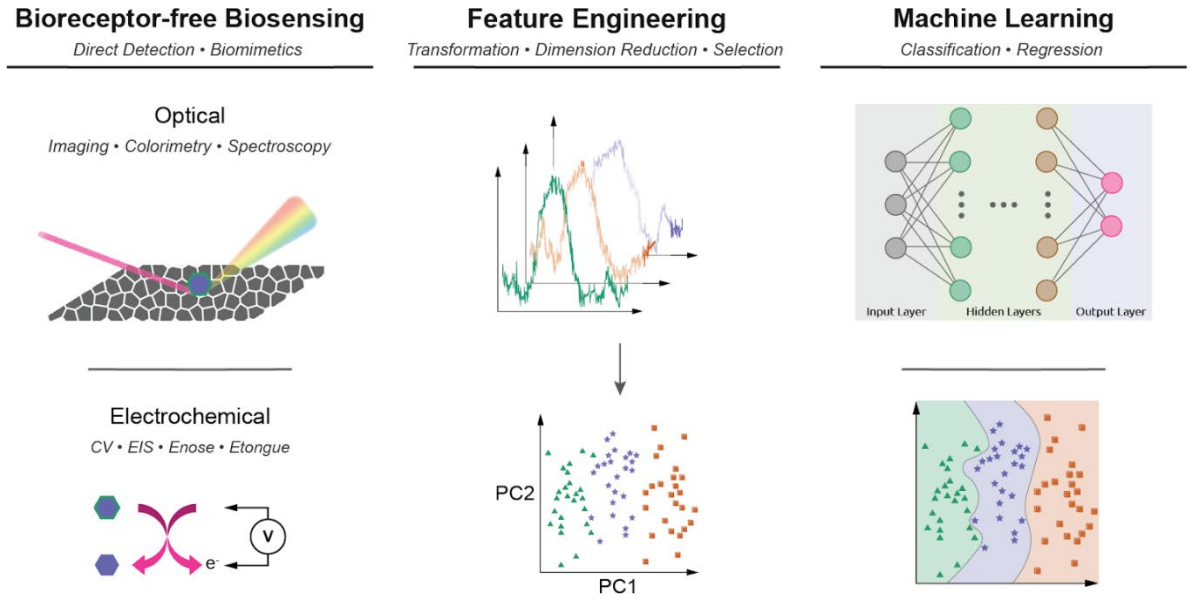

Figure 3

Graphical illustration of machine learning in bioreceptor-free biosensors, including optical and electrochemical methods, feature extraction and engineering, and examples of machine learning methods. Reproduced from (Schackart & Yoon, 2021) with modification under Creative Commons Attributions License.

1.1.3.2. Sequence Classification in Metagenomic Datasets

Microorganisms, including prokaryotes, viruses, and fungi, have been shown to be prevalent and central to all natural ecosystems. These organisms directly and indirectly alter the dynamics of carbon, nitrogen, and other nutrients, with large-scale effects (Docherty & Gutknecht, 2012). Natural microbial communities can be highly diverse, and their interactions complex (Fierer & Jackson, 2006; Hoshino et al., 2020), including interactions with other microorganisms, or hosts such as plants, fungi, and animals including humans (Hoffmann et al., 2016; Singh et al., 2004).

The most abundant biological entities in these communities, and on Earth, are viruses that infect prokaryotes (Ofir & Sorek, 2018). These viruses, called bacteriophages or simply phages, interact with their hosts and can have multiple life cycles. They can actively infect their hosts, causing an active production of viral particles and eventually the death of the host (lytic cycle), or they can integrate their genetic material into the host genome as prophages (lysogenic cycle). Through viral control of host populations, and even host metabolism, phages can play a significant role in ecology and biogeochemical cycling in natural ecosystems (Blazanin & Turner, 2021; Breitbart & Rohwer, 2005).

Culture-based studies are typically not suited for the exploration of complex microbial communities, as a large number of microorganisms can be extremely challenging to culture. If the host cannot be cultured, then phage infecting the host cannot be isolated from viral plaques during infection. Next-generation sequencing technologies have opened new culture-free avenues for studying microbial communities based on DNA isolated directly from an environmental sample called a metagenome. Specifically, scientists can investigate both the taxonomic and functional content of a microbial community, including the phages that are actively infecting these organisms, using Whole Genome Shotgun (WGS) sequencing.

WGS metagenomes are typically composed of sequences coming from any microbes present in the sample. Metagenomes frequently recover phage sequences along with their prokaryotic sequences. For example, in gut metagenomes, it was reported that viral sequences can constitute from 5 to 22% of the total metagenomic content (Shkoporov & Hill, 2019). These sequences come from prophages integrated into their host genomes, as well as DNA from phages actively infecting bacterial cells. Other technical approaches, such as viral-particle enrichment methods allow for a higher proportion of viral sequences in the metagenomes.

To unravel the diversity and role of phages in the natural ecosystem, it is necessary to develop computational approaches that distinguish microbial host from viral sequences in metagenomic datasets. However, identifying viral sequences in large metagenomic datasets is a computational challenge. Because viruses lack a universal gene marker (*e.g.,* 16S rRNA in prokaryotes), have a rapid rate of evolution, and can capture host genes or integrate as prophages in their host genomes, distinguishing viral from bacterial sequences is challenging (Hurwitz et al., 2018; Li et al., 2021). The first bioinformatics methods to identify viruses from metagenomes relied on sequence homology alignment searches against reference genome databases. However, in most ecosystems, these approaches do not capture the total viral diversity. For example, in gut viromes, 75% to 99% of viral reads do not produce significant alignments to any known viral genome (Aggarwala et al., 2017). This can be partially explained by broad under-representation of viral sequences in most genomic databases and the overrepresentation of specific phage groups in these databases. Recent bioinformatic tools leverage ML methods to identify features that signal a phage sequence, and typically allow for a broader recall of previously unknown viral sequences than reference-based approaches. These tools can leverage a large variety of features to inform their models. For instance, VirSorter classifies metagenomic sequences as viral or non-viral by

looking at the enrichment of viral-like and short genes, and depletion of hits to known bacterial proteins and strand shifts (Roux et al., 2015). Other features include gene density (Amgarten et al., 2018), virus hallmark genes (Antipov et al., 2020; J. Guo et al., 2021; Tisza et al., 2021), functional profiling (Kieft et al., 2020; Wommack et al., 2012; T. Zheng et al., 2019), and sequence alignment (Garretto et al., 2019; Jurtz et al., 2016; Zhao et al., 2017). The ML models used include random forest (Amgarten et al., 2018; J. Guo et al., 2021), naive Bayes (Antipov et al., 2020), and neural networks (Kieft et al., 2020).

While these approaches provide more robust classification than simple alignment, their performance is still tightly constrained by database completeness. Novel phages with genes not recorded in the databases will be missed. Additionally, these methods require multiple open reading frames (ORFs) on a sequence to properly classify it. Consequently, short fragmentary metagenomic sequences cannot be classified. Other approaches have been developed which rely only on features of the sequences themselves. These approaches rely on differential *k*-mer (short sequences of length k) frequencies to distinguish bacterial and phage sequences. Indeed, bacteria and phages have distinct distributions of individual *k*-mers, and ML methods can therefore be trained to identify these complex patterns for prediction tasks (Deaton et al., 2017; Ren et al., 2017). While the first tools developed on this concept were using very simple ML models such as logistic regression and k-nearest neighbors, more advanced models such as deep learning neural networks have been recently applied (Ren et al., 2020). The idea that sequence composition itself can be used for this classification task has since been expanded beyond *k*-mers, using the entire sequence. Models employed in this way include CNN (Fang et al., 2019; Tampuu et al., 2019), neural networks with long short-term memory (LSTM) (Auslander et al., 2020; F. Liu et al., 2022), and neural networks which utilize attention (Abdelkareem et al., 2018). These sequence

composition approaches are better able to classify short sequences, since they do not require ORFs, and can potentially generalize better to unknown phages.

1.2. The Need for Improved Standards in Bioscience ML Applications

As discussed previously ML methods are particularly relevant to tackle the current data challenges in many biological fields. They are powerful methods especially suited for detecting patterns from complex and large data.

While the use of ML methods in biological sciences meet current data analysis needs, it also created its own challenges. Perhaps the most critical of these challenges is a lack of reproducibility of ML-based studies (Heil et al., 2021). In particular, the lack of sufficient documentation regarding the training set composition, the training methods and other key software practices can make ML-based studies not reproducible or reusable. Additionally, researchers are often faced with a large array of ML methods and tools for their data analysis. In this context, making an informed choice can be particularly challenging, especially for non-ML experts.

1.2.1 Data, Model Training, and Evaluation Practices

Data handling practices have large impacts on the implementation of ML methods. Often when proper practices are not followed, the result is high but misleading model performance metrics. This may be caused in a number of ways. Data leakage is a common culprit for these inflated performance measures and can be difficult to detect and avoid (Freischlag, 2020). Data leakage occurs when data points in the training set are identical or too closely related to data points in the test set. When this happens, the model performance may be great on the test set, but in reality, the model may be capturing hidden similarities in the training and test sets. This will lead to lower-than-expected performance on real datasets, since the test set is not representative of the different real data the model will encounter. Data leakage may also occur when some processing

is performed on the data before splitting between training and testing sets. For instance, if dimensionality reduction methods are performed on data before splitting, information about the test set is used to inform the model training which again leads to misleading performance measures (Albert, 2021). Test sets must be scrutinized, including assuring that test data was split before any processing, and assessing any potential near duplicates between the training and test sets.

Model selection should be based on relevance to the data, as well as the desired utilization. As advanced deep learning models continue to provide impressive performance on a variety of tasks, it is common to apply these methods on even more tasks. However, considerations must be made before defaulting to these models (Martens et al., 2011). Due to the large number of parameters constituting these models, they require large amounts of training data, which may not be available in some cases (*e.g.,* for patient sets or rare diseases) (Perone & Cohen-Adad, 2019). In many such cases, more simple methods provide robust performance. Importantly, simpler methods are easier to interpret than deep learning methods (Gilpin et al., 2018). Indeed, in many ML applications, the model interpretability is critical since it allows the discovery of the mechanisms and the factors responsible for the modeling decisions.

Finally, it is important that models are evaluated in a reasonable manner. This includes but goes beyond preventing data leakage. Test sets must be representative of data the model will encounter in employment. For instance, when evaluating classifiers, class distributions should be realistic (Paleyes et al., 2022). Additionally, the choice of performance metrics is important. For instance, when evaluating model performance on imbalance data (more representation of certain classes than others) accuracy is inappropriate and may imply high performance when that is not the case. Better metrics for imbalanced data include $F1$ score (Martens et al., 2011).

1.2.2 Reusability and Reproducibility in ML-based Studies

Science, as a methodology, hinges on the ability to test hypotheses experimentally. This means that the hypothesis must be supported by experimental results that can be reproduced. When research findings cannot be reproduced, they must be called into question. Since Ioannidis published the paper "Why Most Published Research Findings Are False", there has been broad concern over the lack of reproducibility in science (Ioannidis, 2005). This phenomenon has been termed the "reproducibility crisis", although this terminology has been criticized (Fanelli, 2018).

The reasons for a poor reproducibility of ML-based studies are multiple, including poor description of methods and assumptions, poor data sharing of the training set or poor code availability and reusability.

To guide ML scientists and developers, Heil *et. al.* proposed a set of standards for reproducibility regarding machine learning in the life sciences (Heil et al., 2021). These standards are ranked in order of increasing reproducibility.

Bronze standard: All data, models, and code used in the analysis are made publicly available. This is the minimum standard for reproducibility. Reproduction may be possible with some effort.

Silver standard: The requirements of the bronze standard still apply to the silver standard. Additionally, all dependencies can be installed with a single command, the key details necessary for reproducing the analysis are documented, and all random components are set to be deterministic. With this standard, reproduction should be easier than with the bronze standard. Analyses that have random components, such as initial parameters in ML models, cannot be reproduced if the randomness is not set to be deterministic.

Gold standard: The gold standard takes reproducibility beyond the silver standard by requiring that the analysis be reproducible with a single command.

Since the general recognition of the problems facing reproducibility, many scientific editors require the release of code and data for any scientific publication (Walters, 2020). While this is an important first step, there is currently no guidance of the code's reusability. Published bioinformatic tools are rarely checked for code completion, code documentation, dependencies inclusions or for any issues or bugs at the time of publication.

In bioinformatics fields and applications, software packages and computational tools are developed specifically for reuse. However, without explicit standards being placed on the code, the reusability of these tools may be compromised, and long-term maintenance of the tools is sometimes difficult to implement.

Better reproducibility standards must be upheld by researchers and enforced by funders and journals to improve reusability of bioinformatic methods and tools. Regarding ML-based studies in the life sciences, the standards laid out by Heil *et. al.* are a starting point for researchers to improve reproducibility.

1.2.3 Benchmarks for Comparisons of Methods

In addition to the reusability and reproducibility of ML methods in the life sciences, it is important to understand how these methods work, how they can be applied, and what their capabilities and limitations are. These comparisons are relevant for users to choose the best suited approaches for their studies, but also to the field at large since it enables identification of current limitations of the available approaches. The speed with which ML applications are being developed and applied makes this important task difficult in practice.

All ML models are designed to perform a specific task and on a specific data type. Many models work under certain assumptions on the data, like the requirement for normally distributed variables discussed previously (Osborne, 2019). Similarly, many models require preprocessing of

the data. For instance, SVM assumes that data are independent and have identical distributions (Hsu et al., 2003). It is important that these requirements be well documented with model distributions and that users understand those requirements. The clear definition of appropriate use cases for ML models can help alleviate improper application, as well as point researchers to similar cases where the models can be applied. To meet this need, "Model Cards for Model Reporting" were proposed (Mitchell et al., 2019). These are documents that accompany trained ML models that describe the intended context of use and provide a benchmarked evaluation on a variety of conditions. While these are intended for documenting trained models, such standards are important for improving the documentation of ML models in research.

Benchmarking, as included in Model Cards (Mitchell et al., 2019), is the evaluation of a method on a standardized evaluation set. Benchmarking efforts are critical to understanding the strengths and limitations of a model, which will inform how it can be employed and how to interpret the results. Benchmarking is most useful when several models are compared on the same standardized set so that they can consistently be compared. The need for benchmarking arises frequently in bioinformatics when several tools designed for the same task are developed, and their relative strengths need to be evaluated. Large-scale benchmarking efforts have been developed for benchmarking tools used for metagenomic read assembly, binning, and taxonomic profiling (Meyer et al., 2019, 2022; Sczyrba et al., 2017). A similar need has also arisen for the detection of phage sequences in metagenomic datasets. Recently, efforts to benchmark phage detection tools have been published, and explore the ability of these tools to correctly classify dsDNA viruses and curate auxiliary metabolic genes (Ho et al., 2022; Pratama et al., 2021). However, the potential impact of parameters such as sequence length, sequencing error, quality of assembly, and phage taxonomy on the tool's performance is not explored.

2. OVERVIEW OF DISSERTATION

Machine learning is particularly suited to the large-scale data produced in the life sciences. With sufficient and high-quality data, robust ML models can be trained to perform tasks untenable using other methods. However, there is a significant need for improved standards regarding the use of ML in the life sciences. First, many studies that leverage ML methods are not fully reproducible. Even published computational tools that are specifically designed for reuse by other scientists can be challenging to reuse and apply. Second, with the rapid development of novel ML methods and models, adequate comparisons among models and computational tools are missing. Significant efforts need to be made to simply understand the landscape of ML in various areas of the life sciences to identify potential uses of developed models. Further, in domains where several models are being used for the same task, benchmarking of those methods is critical for guiding researchers. The aims of this dissertation are to contribute and advance these reusability and reproducibility standards for ML applications in life-sciences. This dissertation contains three manuscripts, which are included as Appendices A, B, and C. Appendix A is a review paper exploring the use of ML for bioreceptor-free biosensing. Appendix B is a benchmarking study comparing bioinformatics tools for classifying phage sequences in metagenomes. Appendix C is an opinion paper describing the efforts made to create a reproducible ML pipeline. The detailed introductions, methods, results, discussion, and conclusions for each chapter are described in the appended materials. The following is a summary of each manuscript.

2.1. Machine Learning Enhances the Performance of Bioreceptor-Free Biosensors

The field of biosensing is broad and with various subfields based on sensing modality. In the various subfields, ML has been applied to many sensor technologies. While within each subfield, there are generally already reviews addressing ML, it is necessary to look at the field as

a whole to guide researchers and identify potential future directions. This review identifies the ML methods that have enabled sensitive and precise biosensors, despite lacking a bioreceptor. In many cases, the methods that are showing the most promising results, based on reported performance, are highlighted. Several opportunities for improvement are also illustrated. For instance, although Enose and Etongue systems both face the challenge of sensor drift, only Enose systems have widely adopted transfer learning to correct for it. By addressing the field as a whole, this review provides guidance to researchers regarding comparisons between candidate models and points them toward future directions. A valuable future direction realized by this study is to assess the use of ML in biosensing on the other standards discussed in this dissertation. For example, to explore the standards of reproducibility individual studies exhibit. This would be an important study direction, to both validate the research findings reported in the review and advance the field with more open science practices.

2.2. A Comprehensive Benchmarking of Metagenomic Phage Detection Tools

Metagenomic approaches have enabled a novel understanding of the complex interactions between phages and their bacterial hosts in natural ecosystems. However, since the majority of phages are unknown, alignment methods are ineffective for detecting the diversity of phage communities in metagenomes. Hence, new computational tools for detecting phage in metagenomes are being rapidly developed, and a critical need has emerged to develop systematic benchmarks for these tools. With an ever-increasing number of computational tools published for the purpose of detecting phage in metagenomes, and many studies drawing inferences on phage communities based on the prediction of those tools, it is critical to understand where these methods perform well, and what are their limitations. This study evaluated the performance of currently available bioinformatic tools and assessed the effect of sequence length, phage taxonomy, low

viral abundance, eukaryotic contamination, sequencing error rate, sequencing technology, and quality of assembly. Strikingly, more than half of the surveyed tools (10/19) could not be installed, run at scale or at all. Of the 9 remaining tools, a wide range of performance could be observed, and different strengths and weaknesses could be identified for each approach.

The results of this effort aim to inform researchers and guide their choice of computational tools for their analytic needs. Further, all of the benchmarking data is released including extensive documentation. Novel phage detection tools are constantly published, and the research community would greatly benefit from comparing these tools' performances on the same standardized dataset.

2.3. A Detailed Case Study on Implementing a Reproducible Machine Learning Enabled Workflow

To address concerns of irreproducibility of studies using ML, standards have been proposed, in particular those developed by Heil *et. al*. While these give useful guidelines, the literature is lacking in specific examples of how these standards can be met in a real research setting. To answer this need, this opinion article describes the rationale and effort behind developing a ML pipeline that meets Heil's Gold Standard of reproducibility. This work is ancillary to a larger project that employs ML to classify scientific articles based on title and abstract, extract key information from the article, as well as perform additional downstream analyses. State-of-art deep learning language models, transformers, are used for the ML tasks. Early in the project, an Open Science Implementation plan was formulated to define the reproducibility, code, and data standards which were to be implemented. This article describes those standards. Discussion includes the rationale behind certain decisions and how they help meet key standards, as well as how they were specifically implemented. We aimed to achieve Heil's gold standard reproducibility through parameterization of code, automation using a workflow

manager, and thorough documentation. Standards which promote robust coding practices were employed, including the use of static code checks, configurability, and automated testing. Data standards are described to facilitate transparency and reproducibility. By providing an example of the efforts made to reach a high level of reproducibility, it is hoped that other researchers may find it helpful in improving their own projects.

3. CONCLUSION

The rise of ML has brought incredible advances to life science research on a variety of problems. The unique ability of these methods to identify and learn patterns from complex and large-scale datasets has provided researchers with powerful new tools. However, the increasingly rapid development and adoption of these methods have brought new challenges. A central challenge of these methods is their reproducibility and reusability. While new methods may be promising, their reusability may be hindered by a variety of problems, such as insufficient documentation. Additionally, researchers are often faced with many potential methods or computational tools for a given task, which makes choosing a method difficult. To address these challenges, it is necessary to improve standards for reproducibility and comparisons of computational methods, both of which have been addressed in this dissertation. A comparison of methods was realized for two domains. First, we assessed the use of ML methods for bioreceptor-free biosensing, pointing out patterns of use and areas where certain methods could be useful. Second, we conducted a comprehensive benchmarking study on tools designed to detect phage sequences in metagenomes providing valuable insights into the various tools' strengths and limitations. To encourage further benchmarking efforts, the datasets used in this study are made publicly available with detailed documentation. The standards of reproducibility were also addressed for two domains. During the benchmarking study, all available phage identification tools

were evaluated for reusability. For tools that we were not able to use, the reasons for this were documented (*e.g.,* insufficient documentation, hard-coded paths, and inability to scale to real datasets). Ideally, developers of new tools can use this to guide decisions to enable greater reusability. Finally, we provided a detailed description of the efforts and considerations that were made to produce a reproducible ML workflow including explicit reproducibility, code quality, and data standards and how they were attained. Overall, this dissertation has aimed to advance standards for machine learning in the life sciences and has done so through efforts to provide comparisons and benchmarking of methods and evaluate and extend their reproducibility and reusability.

4. REFERENCES

Abdelkareem, A. O., Khalil, M. I., Elaraby, M., Abbas, H., & Elbehery, A. H. A. (2018). VirNet: Deep attention model for viral reads identification. *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 623–626. https://doi.org/10.1109/ICCES.2018.8639400

Aggarwala, V., Liang, G., & Bushman, F. D. (2017). Viral communities of the human gut: Metagenomic analysis of composition and dynamics. *Mobile DNA*, *8*(1), 12. https://doi.org/10.1186/s13100-017-0095-y

Agrawal, S. K. (2021, July 20). Evaluation Metrics For Classification Model | Classification Model Metrics. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

Ahmed, A., Rushworth, J. V., Hirst, N. A., & Millner, P. A. (2014). Biosensors for Whole-Cell Bacterial Detection. *Clinical Microbiology Reviews*, *27*(3), 631–646. https://doi.org/10.1128/CMR.00120-13

Albert. (2021, March 9). *Data Leakage Examples in Machine Learning*. https://blog.albertkuo.me/post/data-leakage/

Amgarten, D., Braga, L. P. P., da Silva, A. M., & Setubal, J. C. (2018). MARVEL, a Tool for Prediction of Bacteriophage Sequences in Metagenomic Bins. *Frontiers in Genetics*, *9*. https://www.frontiersin.org/articles/10.3389/fgene.2018.00304

Antipov, D., Raiko, M., Lapidus, A., & Pevzner, P. A. (2020). MetaviralSPAdes: Assembly of viruses from metagenomic data. *Bioinformatics*, *36*(14), 4126–4129. https://doi.org/10.1093/bioinformatics/btaa490

Arnaud, É., Elbattah, M., Gignon, M., & Dequen, G. (2021). NLP-Based Prediction of Medical Specialties at Hospital Admission Using Triage Notes. *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, 548–553. https://doi.org/10.1109/ICHI52183.2021.00103

Auslander, N., Gussow, A. B., Benler, S., Wolf, Y. I., & Koonin, E. V. (2020). Seeker: Alignment-free identification of bacteriophage genomes by deep learning. *Nucleic Acids Research*, *48*(21), e121–e121. https://doi.org/10.1093/nar/gkaa856

Bachmanov, A., Bosak, N., Lin, C., Matsumoto, I., Ohmoto, M., R. Reed, D., & M. Nelson, T. (2014). Genetics of Taste Receptors. *Current Pharmaceutical Design*, *20*(16), 2669–2683.

Blazanin, M., & Turner, P. E. (2021). Community context matters for bacteria-phage ecology and evolution. *The ISME Journal*, *15*(11), Article 11. https://doi.org/10.1038/s41396-021-01012-x

Borhani, N., Bower, A. J., Boppart, S. A., & Psaltis, D. (2019). Digital staining through the application of deep neural networks to multi-modal multi-photon microscopy. *Biomedical Optics Express*, *10*(3), 1339–1350. https://doi.org/10.1364/BOE.10.001339

Botchkarev, A. (2019). Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *Interdisciplinary Journal of Information, Knowledge, and Management*, *14*, 045–076. https://doi.org/10.28945/4184

Breitbart, M., & Rohwer, F. (2005). Here a virus, there a virus, everywhere the same virus? *Trends in Microbiology*, *13*(6), 278–284. https://doi.org/10.1016/j.tim.2005.04.003

Brownlee, J. (2017, July 27). Why One-Hot Encode Data in Machine Learning? *MachineLearningMastery.Com*. https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

Chen, X., Lan, J., Liu, Y., Li, L., Yan, L., Xia, Y., Wu, F., Li, C., Li, S., & Chen, J. (2018). A paper-supported aptasensor based on upconversion luminescence resonance energy transfer for the accessible determination of exosomes. *Biosensors and Bioelectronics*, *102*, 582–588. https://doi.org/10.1016/j.bios.2017.12.012

Chen, X., Nguyen, T. H. D., Gu, L., & Lin, M. (2017). Use of Standing Gold Nanorods for Detection of Malachite Green and Crystal Violet in Fish by SERS. *Journal of Food Science*, *82*(7), 1640–1646. https://doi.org/10.1111/1750-3841.13766

Cunningham, P. (2008). Dimension Reduction. In M. Cord & P. Cunningham (Eds.), *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval* (pp. 91–112). Springer. https://doi.org/10.1007/978-3-540-75171-7_4

Deaton, J., Yu, F. B., & Quake, S. R. (2017). *PhaMers identifies novel bacteriophage sequences from thermophilic hot springs* (p. 169672). bioRxiv. https://doi.org/10.1101/169672

Docherty, K. M., & Gutknecht, J. L. M. (2012). The role of environmental microorganisms in ecosystem responses to global change: Current state of research and future outlooks. *Biogeochemistry*, *109*(1), 1–6. https://doi.org/10.1007/s10533-011-9614-y

Dunker, S., Motivans, E., Rakosy, D., Boho, D., Mäder, P., Hornick, T., & Knight, T. M. (2021). Pollen analysis using multispectral imaging flow cytometry and deep learning. *New Phytologist*, *229*(1), 593–606. https://doi.org/10.1111/nph.16882

Durante, G., Becari, W., Lima, F. A. S., & Peres, H. E. M. (2016). Electrical Impedance Sensor for Real-Time Detection of Bovine Milk Adulteration. *IEEE Sensors Journal*, *16*(4), 861–865. https://doi.org/10.1109/JSEN.2015.2494624

Edachana, R. P., Kumaresan, A., Balasubramanian, V., Thiagarajan, R., Nair, B. G., & Thekkedath Gopalakrishnan, S. B. (2019). Paper-based device for the colorimetric assay of bilirubin based on in-situ formation of gold nanoparticles. *Microchimica Acta*, *187*(1), 60. https://doi.org/10.1007/s00604-019-4051-z

Fanelli, D. (2018). Is science really facing a reproducibility crisis, and do we need it to? *Proceedings of the National Academy of Sciences*, *115*(11), 2628–2631. https://doi.org/10.1073/pnas.1708272114

Fang, Z., Tan, J., Wu, S., Li, M., Xu, C., Xie, Z., & Zhu, H. (2019). PPR-Meta: A tool for identifying phages and plasmids from metagenomic fragments using deep learning. *GigaScience*, *8*(6), giz066. https://doi.org/10.1093/gigascience/giz066

Fierer, N., & Jackson, R. B. (2006). The diversity and biogeography of soil bacterial communities. *Proceedings of the National Academy of Sciences*, *103*(3), 626–631. https://doi.org/10.1073/pnas.0507535103

Freischlag, C. (2020, May 19). *How data leakage affects machine learning models in practice*. Medium. https://towardsdatascience.com/how-data-leakage-affects-machine-learning-models-in-practice-f448be6080d0

Garretto, A., Hatzopoulos, T., & Putonti, C. (2019). virMine: Automated detection of viral sequences from complex metagenomic samples. *PeerJ*, *7*, e6695. https://doi.org/10.7717/peerj.6695

Gavrilescu, M., Demnerová, K., Aamand, J., Agathos, S., & Fava, F. (2015). Emerging pollutants in the environment: Present and future challenges in biomonitoring, ecological risks and bioremediation. *New Biotechnology*, *32*(1), 147–156. https://doi.org/10.1016/j.nbt.2014.01.001

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining Explanations: An Overview of Interpretability of Machine Learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 80–89. https://doi.org/10.1109/DSAA.2018.00018

Guo, J., Bolduc, B., Zayed, A. A., Varsani, A., Dominguez-Huerta, G., Delmont, T. O., Pratama, A. A., Gazitúa, M. C., Vik, D., Sullivan, M. B., & Roux, S. (2021). VirSorter2: A multi-

classifier, expert-guided approach to detect diverse DNA and RNA viruses. *Microbiome*, *9*(1), 37. https://doi.org/10.1186/s40168-020-00990-y

Guo, X. (2012). Surface plasmon resonance based biosensor technique: A review. *Journal of Biophotonics*, *5*(7), 483–501. https://doi.org/10.1002/jbio.201200015

Hasib, K. M., Iqbal, M. S., Shah, F. M., Mahmud, J. A., Popel, M. H., Showrov, M. I. H., Ahmed, S., & Rahman, O. (2020). A Survey of Methods for Managing the Classification and Solution of Data Imbalance Problem. *Journal of Computer Science*, *16*(11), 1546–1557. https://doi.org/10.3844/jcssp.2020.1546.1557

Hassoun, M., Rüger, J., Kirchberger-Tolstik, T., Schie, I. W., Henkel, T., Weber, K., Cialla-May, D., Krafft, C., & Popp, J. (2018). A droplet-based microfluidic chip as a platform for leukemia cell lysate identification using surface-enhanced Raman scattering. *Analytical and Bioanalytical Chemistry*, *410*(3), 999–1006. https://doi.org/10.1007/s00216-017-0609-y

Hastie, T., Friedman, J., & Tibshirani, R. (2001). Model Assessment and Selection. In T. Hastie, J. Friedman, & R. Tibshirani (Eds.), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (pp. 193–224). Springer. https://doi.org/10.1007/978-0-387-21606-5_7

Heil, B. J., Hoffman, M. M., Markowetz, F., Lee, S.-I., Greene, C. S., & Hicks, S. C. (2021). Reproducibility standards for machine learning in the life sciences. *Nature Methods*, *18*(10), Article 10. https://doi.org/10.1038/s41592-021-01256-7

Heinze, B. C., & Yoon, J.-Y. (2011). Nanoparticle immunoagglutination Rayleigh scatter assay to complement microparticle immunoagglutination Mie scatter assay in a microfluidic

device. *Colloids and Surfaces B: Biointerfaces*, *85*(2), 168–173. https://doi.org/10.1016/j.colsurfb.2011.02.024

Helwan, A., Idoko, J. B., & Abiyev, R. H. (2017). Machine learning techniques for classification of breast tissue. *Procedia Computer Science*, *120*, 402–410. https://doi.org/10.1016/j.procs.2017.11.256

Hidi, I. J., Jahn, M., Pletz, M. W., Weber, K., Cialla-May, D., & Popp, J. (2016a). Toward Levofloxacin Monitoring in Human Urine Samples by Employing the LoC-SERS Technique. *The Journal of Physical Chemistry C*, *120*(37), 20613–20623. https://doi.org/10.1021/acs.jpcc.6b01005

Hidi, I. J., Jahn, M., Pletz, M. W., Weber, K., Cialla-May, D., & Popp, J. (2016b). Toward Levofloxacin Monitoring in Human Urine Samples by Employing the LoC-SERS Technique. *The Journal of Physical Chemistry C*, *120*(37), 20613–20623. https://doi.org/10.1021/acs.jpcc.6b01005

Ho, S. F. S., Wheeler, N., Millard, A. D., & Schaik, W. van. (2022). *Gauge your phage: Benchmarking of bacteriophage identification tools in metagenomic sequencing data* (p. 2021.04.12.438782). bioRxiv. https://doi.org/10.1101/2021.04.12.438782

Hock, B. (1997). Antibodies for immunosensors a review. *Analytica Chimica Acta*, *347*(1), 177–186. https://doi.org/10.1016/S0003-2670(97)00167-0

Hoffmann, A. R., Proctor, L. M., Surette, M. G., & Suchodolski, J. S. (2016). The Microbiome: The Trillions of Microorganisms That Maintain Health and Cause Disease in Humans and Companion Animals. *Veterinary Pathology*, *53*(1), 10–21. https://doi.org/10.1177/0300985815595517

Hoshino, T., Doi, H., Uramoto, G.-I., Wörmer, L., Adhikari, R. R., Xiao, N., Morono, Y., D'Hondt, S., Hinrichs, K.-U., & Inagaki, F. (2020). Global diversity of microbial communities in marine sediment. *Proceedings of the National Academy of Sciences*, *117*(44), 27587–27597. https://doi.org/10.1073/pnas.1919139117

Hotel, O., Poli, J.-P., Mer-Calfati, C., Scorsone, E., & Saada, S. (2018). A review of algorithms for SAW sensors e-nose based volatile compound identification. *Sensors and Actuators B: Chemical*, *255*, 2472–2482. https://doi.org/10.1016/j.snb.2017.09.040

Hsu, C.-W., Chang, C.-C., Lin, C.-J., & others. (2003). *A practical guide to support vector classification*. Taipei, Taiwan.

Hurwitz, B. L., Ponsero, A., Thornton, J., & U'Ren, J. M. (2018). Phage hunters: Computational strategies for finding phages in large-scale 'omics datasets. *Virus Research*, *244*, 110–115. https://doi.org/10.1016/j.virusres.2017.10.019

Ioannidis, J. P. A. (2005). Why Most Published Research Findings Are False. *PLOS Medicine*, *2*(8), e124. https://doi.org/10.1371/journal.pmed.0020124

Islam, M., Wahid, K., & Dinh, A. (2018). *Assessment of Ripening Degree of Avocado by Electrical Impedance Spectroscopy and Support Vector Machine* [Research Article]. Journal of Food Quality; Hindawi. https://doi.org/10.1155/2018/4706147

Izenman, A. J. (2008). Linear Discriminant Analysis. In A. J. Izenman (Ed.), *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning* (pp. 237–280). Springer. https://doi.org/10.1007/978-0-387-78189-1_8

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), Article 7873. https://doi.org/10.1038/s41586-021-03819-2

Jurtz, V. I., Villarroel, J., Lund, O., Larsen, M. V., & Nielsen, M. (2016). MetaPhinder—Identifying Bacteriophage Sequences in Metagenomic Data Sets. *PLOS ONE*, *11*(9), e0163111. https://doi.org/10.1371/journal.pone.0163111

Kämmer, E., Olschewski, K., Stöckel, S., Rösch, P., Weber, K., Cialla-May, D., Bocklitz, T., & Popp, J. (2015). Quantitative SERS studies by combining LOC-SERS with the standard addition method. *Analytical and Bioanalytical Chemistry*, *407*(29), 8925–8929. https://doi.org/10.1007/s00216-015-9045-z

Kieft, K., Zhou, Z., & Anantharaman, K. (2020). VIBRANT: Automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences. *Microbiome*, *8*(1), 90. https://doi.org/10.1186/s40168-020-00867-0

Koohy, H. (2018). The rise and fall of machine learning methods in biomedical research. *F1000Research*, *6*, 2012. https://doi.org/10.12688/f1000research.13016.2

Leon-Medina, J. X., Pineda-Muñoz, W. A., & Burgos, D. A. T. (2020). Joint Distribution Adaptation for Drift Correction in Electronic Nose Type Sensor Arrays. *IEEE Access*, *8*, 134413–134421. https://doi.org/10.1109/ACCESS.2020.3010711

Li, Y., Handley, S. A., & Baldridge, M. T. (2021). The dark side of the gut: Virome–host interactions in intestinal homeostasis and disease. *Journal of Experimental Medicine*, *218*(5), e20201044. https://doi.org/10.1084/jem.20201044

Lim, Y. C., Kouzani, A. Z., & Duan, W. (2010). Aptasensors: A Review. *Journal of Biomedical Nanotechnology*, *6*(2), 93–105. https://doi.org/doi:10.1166/jbn.2010.1103

Lin, B., Yu, Y., Cao, Y., Guo, M., Zhu, D., Dai, J., & Zheng, M. (2018). Point-of-care testing for streptomycin based on aptamer recognizing and digital image colorimetry by smartphone. *Biosensors and Bioelectronics*, *100*, 482–489. https://doi.org/10.1016/j.bios.2017.09.028

Liu, B., Zeng, X., Tian, F., Zhang, S., & Zhao, L. (2019). Domain Transfer Broad Learning System for Long-Term Drift Compensation in Electronic Nose Systems. *IEEE Access*, *7*, 143947–143959. https://doi.org/10.1109/ACCESS.2019.2943188

Liu, F., Miao, Y., Liu, Y., & Hou, T. (2022). RNN-VirSeeker: A Deep Learning Method for Identification of Short Viral Sequences From Metagenomes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *19*(3), 1840–1849. https://doi.org/10.1109/TCBB.2020.3044575

Lotfollahi, M., Berisha, S., Daeinejad, D., & Mayerich, D. (2019). Digital Staining of High-Definition Fourier Transform Infrared (FT-IR) Images Using Deep Learning. *Applied Spectroscopy*, *73*(5), 556–564.

Maeda, T. (2018). *Technical note: How to rationally compare the performances of different machine learning models?* (e26714v1). PeerJ Inc. https://doi.org/10.7287/peerj.preprints.26714v1

Mahmudunnabi, R., Zerin Farhana, F., Kashaninejad, N., H. Firoz, S., Shim, Y.-B., & A. Shiddiky, M. J. (2020). Nanozyme-based electrochemical biosensors for disease

biomarker detection. *Analyst*, *145*(13), 4398–4420. https://doi.org/10.1039/D0AN00558D

Martens, D., Vanthienen, J., Verbeke, W., & Baesens, B. (2011). Performance of classification models from a user perspective. *Decision Support Systems*, *51*(4), 782–793. https://doi.org/10.1016/j.dss.2011.01.013

Massah, J., & Asefpour Vakilian, K. (2019). An intelligent portable biosensor for fast and accurate nitrate determination using cyclic voltammetry. *Biosystems Engineering*, *177*, 49–58. https://doi.org/10.1016/j.biosystemseng.2018.09.007

Meyer, F., Bremges, A., Belmann, P., Janssen, S., McHardy, A. C., & Koslicki, D. (2019). Assessing taxonomic metagenome profilers with OPAL. *Genome Biology*, *20*(1), 51. https://doi.org/10.1186/s13059-019-1646-y

Meyer, F., Fritz, A., Deng, Z.-L., Koslicki, D., Lesker, T. R., Gurevich, A., Robertson, G., Alser, M., Antipov, D., Beghini, F., Bertrand, D., Brito, J. J., Brown, C. T., Buchmann, J., Buluç, A., Chen, B., Chikhi, R., Clausen, P. T. L. C., Cristian, A., … McHardy, A. C. (2022). Critical Assessment of Metagenome Interpretation: The second round of challenges. *Nature Methods*, *19*(4), Article 4. https://doi.org/10.1038/s41592-022-01431-4

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model Cards for Model Reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229. https://doi.org/10.1145/3287560.3287596

Mühlig, A., Bocklitz, T., Labugger, I., Dees, S., Henk, S., Richter, E., Andres, S., Merker, M., Stöckel, S., Weber, K., Cialla-May, D., & Popp, J. (2016). LOC-SERS: A Promising

Closed System for the Identification of Mycobacteria. *Analytical Chemistry*, *88*(16), 7998–8004. https://doi.org/10.1021/acs.analchem.6b01152

Murphy, E. K., Mahara, A., Khan, S., Hyams, E. S., Schned, A. R., Pettus, J., & Halter, R. J. (2017). Comparative study of separation betweenex vivoprostatic malignant and benign tissue using electrical impedance spectroscopy and electrical impedance tomography. *Physiological Measurement*, *38*(6), 1242–1261. https://doi.org/10.1088/1361-6579/aa660e

Nguyen, C. Q., Thrift, W. J., Bhattacharjee, A., Ranjbar, S., Gallagher, T., Darvishzadeh-Varcheie, M., Sanderson, R. N., Capolino, F., Whiteson, K., Baldi, P., Hochbaum, A. I., & Ragan, R. (2018). Longitudinal Monitoring of Biofilm Formation via Robust Surface-Enhanced Raman Scattering Quantification of Pseudomonas aeruginosa-Produced Metabolites. *ACS Applied Materials & Interfaces*, *10*(15), 12364–12373. https://doi.org/10.1021/acsami.7b18592

Nguyen, H. H., Lee, S. H., Lee, U. J., Fermin, C. D., & Kim, M. (2019). Immobilized Enzymes in Biosensor Applications. *Materials*, *12*(1), Article 1. https://doi.org/10.3390/ma12010121

Ofir, G., & Sorek, R. (2018). Contemporary Phage Biology: From Classic Models to New Insights. *Cell*, *172*(6), 1260–1270. https://doi.org/10.1016/j.cell.2017.10.045

Osborne, J. (2019). Notes on the use of data transformations. *Practical Assessment, Research, and Evaluation*, *8*(1). https://doi.org/10.7275/4vng-5608

Ozer, T., & Henry, C. S. (2021). Review—Recent Advances in Sensor Arrays for the Simultaneous Electrochemical Detection of Multiple Analytes. *Journal of The Electrochemical Society*, *168*(5), 057507. https://doi.org/10.1149/1945-7111/abfc9f

Paleyes, A., Urma, R.-G., & Lawrence, N. D. (2022). Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Computing Surveys*. https://doi.org/10.1145/3533378

Perone, C. S., & Cohen-Adad, J. (2019). Promises and limitations of deep learning for medical image segmentation. *Journal of Medical Artificial Intelligence*, *2*(0), Article 0. https://doi.org/10.21037/jmai.2019.01.01

Picard, R. R., & Cook, R. D. (1984). Cross-Validation of Regression Models. *Journal of the American Statistical Association*, *79*(387), 575–583. https://doi.org/10.1080/01621459.1984.10478083

Pratama, A. A., Bolduc, B., Zayed, A. A., Zhong, Z.-P., Guo, J., Vik, D. R., Gazitúa, M. C., Wainaina, J. M., Roux, S., & Sullivan, M. B. (2021). Expanding standards in viromics: In silico evaluation of dsDNA viral genome identification, classification, and auxiliary metabolic gene curation. *PeerJ*, *9*, e11447. https://doi.org/10.7717/peerj.11447

Rajoub, B. (2020). Chapter 3—Supervised and unsupervised learning. In W. Zgallai (Ed.), *Biomedical Signal Processing and Artificial Intelligence in Healthcare* (pp. 51–89). Academic Press. https://doi.org/10.1016/B978-0-12-818946-7.00003-2

Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A., & Sun, F. (2017). VirFinder: A novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, *5*(1), 69. https://doi.org/10.1186/s40168-017-0283-5

Ren, J., Song, K., Deng, C., Ahlgren, N. A., Fuhrman, J. A., Li, Y., Xie, X., Poplin, R., & Sun, F. (2020). Identifying viruses from metagenomic data using deep learning. *Quantitative Biology*, *8*(1), 64–77. https://doi.org/10.1007/s40484-019-0187-4

Rivenson, Y., Liu, T., Wei, Z., Zhang, Y., de Haan, K., & Ozcan, A. (2019). PhaseStain: The digital staining of label-free quantitative phase microscopy images using deep learning. *Light: Science & Applications*, *8*(1), Article 1. https://doi.org/10.1038/s41377-019-0129-y

Rivenson, Y., Wu, Y., & Ozcan, A. (2019). Deep learning in holography and coherent imaging. *Light: Science & Applications*, *8*(1), Article 1. https://doi.org/10.1038/s41377-019-0196-0

Roux, S., Enault, F., Hurwitz, B. L., & Sullivan, M. B. (2015). VirSorter: Mining viral signal from microbial genomic data. *PeerJ*, *3*, e985. https://doi.org/10.7717/peerj.985

Roy, M., Seo, D., Oh, S., Yang, J.-W., & Seo, S. (2017). A review of recent progress in lens-free imaging and sensing. *Biosensors and Bioelectronics*, *88*, 130–143. https://doi.org/10.1016/j.bios.2016.07.115

Sanaeifar, A., ZakiDizaji, H., Jafari, A., & Guardia, M. de la. (2017). Early detection of contamination and defect in foodstuffs by electronic nose: A review. *TrAC Trends in Analytical Chemistry*, *97*, 257–271. https://doi.org/10.1016/j.trac.2017.09.014

Schackart, K. E., & Yoon, J.-Y. (2021). Machine Learning Enhances the Performance of Bioreceptor-Free Biosensors. *Sensors*, *21*(16). https://doi.org/10.3390/s21165519

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., Bremges, A., Fritz, A., Garrido-Oter, R., Jørgensen, T. S., Shapiro, N., Blood, P. D., Gurevich, A., Bai, Y., Turaev, D., … McHardy, A. C. (2017).

Critical Assessment of Metagenome Interpretation—A benchmark of metagenomics software. *Nature Methods*, *14*(11), Article 11. https://doi.org/10.1038/nmeth.4458

Sheng, Y., Qian, W., Huang, J., Wu, B., Yang, J., Xue, T., Ge, Y., & Wen, Y. (2019). Electrochemical detection combined with machine learning for intelligent sensing of maleic hydrazide by using carboxylated PEDOT modified with copper nanoparticles. *Microchimica Acta*, *186*(8), 543. https://doi.org/10.1007/s00604-019-3652-x

Shkoporov, A. N., & Hill, C. (2019). Bacteriophages of the Human Gut: The "Known Unknown" of the Microbiome. *Cell Host & Microbe*, *25*(2), 195–209. https://doi.org/10.1016/j.chom.2019.01.017

Singh, B. K., Millard, P., Whiteley, A. S., & Murrell, J. C. (2004). Unravelling rhizosphere–microbial interactions: Opportunities and limitations. *Trends in Microbiology*, *12*(8), 386–393. https://doi.org/10.1016/j.tim.2004.06.008

Song, E., Yu, M., Wang, Y., Hu, W., Cheng, D., Swihart, M. T., & Song, Y. (2015). Multi-color quantum dot-based fluorescence immunoassay array for simultaneous visual detection of multiple antibiotic residues in milk. *Biosensors and Bioelectronics*, *72*, 320–325. https://doi.org/10.1016/j.bios.2015.05.018

Sun, H., Lv, G., Mo, J., Lv, X., Du, G., & Liu, Y. (2019). Application of KPCA combined with SVM in Raman spectral discrimination. *Optik*, *184*, 214–219. https://doi.org/10.1016/j.ijleo.2019.02.126

Tampuu, A., Bzhalava, Z., Dillner, J., & Vicente, R. (2019). ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. *PLOS ONE*, *14*(9), e0222271. https://doi.org/10.1371/journal.pone.0222271

Tan, J., & Xu, J. (2020). Applications of electronic nose (e-nose) and electronic tongue (e-tongue) in food quality-related properties determination: A review. *Artificial Intelligence in Agriculture*, *4*, 104–115. https://doi.org/10.1016/j.aiia.2020.06.003

Tisza, M. J., Belford, A. K., Domínguez-Huerta, G., Bolduc, B., & Buck, C. B. (2021). Cenote-Taker 2 democratizes virus discovery and sequence annotation. *Virus Evolution*, *7*(1), veaa100. https://doi.org/10.1093/ve/veaa100

Tomczak, A., Ilic, S., Marquardt, G., Engel, T., Forster, F., Navab, N., & Albarqouni, S. (2020). Multi-task multi-domain learning for digital staining and classification of leukocytes. *IEEE Transactions on Medical Imaging*, 1–1. https://doi.org/10.1109/TMI.2020.3046334

Tonezzer, M., Le, D. T. T., Iannotta, S., & Van Hieu, N. (2018). Selective discrimination of hazardous gases using one single metal oxide resistive sensor. *Sensors and Actuators B: Chemical*, *277*, 121–128. https://doi.org/10.1016/j.snb.2018.08.103

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Walters, W. P. (2020). Code Sharing in the Open Science Era. *Journal of Chemical Information and Modeling*, *60*(10), 4417–4420. https://doi.org/10.1021/acs.jcim.0c01000

Wetterstrand, K. A. (2021, November 1). *The Cost of Sequencing a Human Genome*. Genome.Gov. https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

Wide, L., & Gemzell, C. A. (1960). An immunological pregnancy test. *European Journal of Endocrinology*, *35*(II), 261–267.

Winship, C., & Mare, R. D. (1984). Regression Models with Ordinal Variables. *American Sociological Review*, *49*(4), 512–525. https://doi.org/10.2307/2095465

Wommack, K. E., Bhavsar, J., Polson, S. W., Chen, J., Dumas, M., Srinivasiah, S., Furman, M., Jamindar, S., & Nasko, D. J. (2012). VIROME: A standard operating procedure for analysis of viral metagenome sequences. *Standards in Genomic Sciences*, *6*(3), Article 3. https://doi.org/10.4056/sigs.2945050

Wu, Y., Ray, A., Wei, Q., Feizi, A., Tong, X., Chen, E., Luo, Y., & Ozcan, A. (2019). Deep Learning Enables High-Throughput Analysis of Particle-Aggregation-Based Biosensors Imaged Using Holography. *ACS Photonics*, *6*(2), 294–301. https://doi.org/10.1021/acsphotonics.8b01479

Yang, T., Guo, X., Wang, H., Fu, S., wen, Y., & Yang, H. (2015). Magnetically optimized SERS assay for rapid detection of trace drug-related biomarkers in saliva and fingerprints. *Biosensors and Bioelectronics*, *68*, 350–357. https://doi.org/10.1016/j.bios.2015.01.021

Yang, Y., Liu, H., & Gu, Y. (2020). A Model Transfer Learning Framework With Back-Propagation Neural Network for Wine and Chinese Liquor Detection by Electronic Nose. *IEEE Access*, *8*, 105278–105285. https://doi.org/10.1109/ACCESS.2020.2999591

Yang, Z., Miao, N., Zhang, X., Li, Q., Wang, Z., Li, C., Sun, X., & Lan, Y. (2021). Employment of an electronic tongue combined with deep learning and transfer learning for discriminating the storage time of Pu-erh tea. *Food Control*, *121*, 107608. https://doi.org/10.1016/j.foodcont.2020.107608

Yi, R., Yan, J., Shi, D., Tian, Y., Chen, F., Wang, Z., & Duan, S. (2021). Improving the performance of drifted/shifted electronic nose systems by cross-domain transfer using

common transfer samples. *Sensors and Actuators B: Chemical*, *329*, 129162. https://doi.org/10.1016/j.snb.2020.129162

Y. Mutlu, A., Kılıç, V., Kocakuşak Özdemir, G., Bayram, A., Horzum, N., & E. Solmaz, M. (2017). Smartphone-based colorimetric detection via machine learning. *Analyst*, *142*(13), 2434–2441. https://doi.org/10.1039/C7AN00741H

Yoon, J.-Y. (2016). Introduction to Biosensors: From Electric Circuits to Immunosensors. In J.-Y. Yoon (Ed.), *Introduction to Biosensors: From Electric Circuits to Immunosensors* (pp. 1–15). Springer International Publishing. https://doi.org/10.1007/978-3-319-27413-3_1

Zhang, D., Huang, L., Liu, B., Ni, H., Sun, L., Su, E., Chen, H., Gu, Z., & Zhao, X. (2018). Quantitative and ultrasensitive detection of multiplex cardiac biomarkers in lateral flow assay with core-shell SERS nanotags. *Biosensors and Bioelectronics*, *106*, 204–211. https://doi.org/10.1016/j.bios.2018.01.062

Zhao, G., Wu, G., Lim, E. S., Droit, L., Krishnamurthy, S., Barouch, D. H., Virgin, H. W., & Wang, D. (2017). VirusSeeker, a computational pipeline for virus discovery and virome composition analysis. *Virology*, *503*, 21–30. https://doi.org/10.1016/j.virol.2017.01.005

Zheng, T., Li, J., Ni, Y., Kang, K., Misiakou, M.-A., Imamovic, L., Chow, B. K. C., Rode, A. A., Bytzer, P., Sommer, M., & Panagiotou, G. (2019). Mining, analyzing, and integrating viral signals from metagenomic data. *Microbiome*, *7*(1), 42. https://doi.org/10.1186/s40168-019-0657-y

Zheng, X., Lv, G., Du, G., Zhai, Z., Mo, J., & Lv, X. (2018). Rapid and Low-Cost Detection of Thyroid Dysfunction Using Raman Spectroscopy and an Improved Support Vector Machine. *IEEE Photonics Journal*, *10*(6), 1–12. https://doi.org/10.1109/JPHOT.2018.2876686

Zhu, X., & Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *3*(1), 1–130. https://doi.org/10.2200/S00196ED1V01Y200906AIM006

APPENDIX A

Machine Learning Enhances the Performance of Bioreceptor-Free Biosensors

Kenneth E Schackart III[1], and Jeong-Yeol Yoon[1,2]

[1]Department of Biosystems Engineering, The University of Arizona, Tucson, Arizona 85721, United States. [2]Department of Biomedical Engineering, The University of Arizona, Tucson, Arizona 85721, United States.

Correspondence and requests for materials should be addressed to J.-Y.Y. (email: jyyoon@email.arizona.edu)

Recieved 15 July 2021; Revised 9 August 2021; Accepted 13 August 2021; Published 17 August 2021

Sensors **2021**, *21*(16), 5519; https://doi.org/10.3390/s21165519

Abstract

Since their inception, biosensors have frequently employed simple regression models to calculate analyte composition based on the biosensor's signal magnitude. Traditionally, bioreceptors provide excellent sensitivity and specificity to the biosensor. Increasingly, however, bioreceptor-free biosensors have been developed for a wide range of applications. Without a bioreceptor, maintaining strong specificity and a low limit of detection have become the major challenge. Machine learning (ML) has been introduced to improve the performance of these biosensors, effectively replacing the bioreceptor with modeling to gain specificity. Here, we present how ML has been used to enhance the performance of these bioreceptor-free biosensors. Particularly, we discuss how ML has been used for imaging, Enose and Etongue, and surface-enhanced Raman spectroscopy (SERS) biosensors. Notably, principal component analysis (PCA) combined with support vector machine (SVM) and various artificial neural network (ANN) algorithms have shown out-standing performance in a variety of tasks. We anticipate that ML will continue to improve the performance of bioreceptor-free biosensors, especially with the prospects of sharing trained models and cloud computing for mobile computation. To facilitate this, the biosensing community would benefit from increased contributions to open-access data repositories for biosensor data.

Keywords: label-free biosensor; machine learning; support vector machine; artificial neural network; principal component analysis

1. Introduction

The field of biosensing has exploded into nearly all areas of research, from medical applications [1] to environmental monitoring [2]. Some of the greatest appeals of biosensors are their specificity and sensitivity. These properties are primarily due to bio-receptors, which are

selected for their inherent specificities such as enzymes [3], anti-bodies [4], and aptamers [5]. However, the very aspect that makes biosensors so specific and sensitive can also limit the sensor stability due to the degradation of the bioreceptor [6]. Additionally, as the bioreceptor is specific to an individual analyte, the particular sensor's scope is limited to the specific analyte to which the bioreceptor can bind.

To obviate these issues, many nature-inspired sensors have emerged that are bio-receptor-free. Some of the most notable examples that have made great progress include the electronic nose (Enose) [7–11] and electronic tongue (Etongue) [12–16]. Additionally, surface enhanced Raman spectroscopy (SERS)-based sensors have demonstrated in-credible chemosensing ability [17–21]. Without a bioreceptor, however, there is the risk of significantly compromised biosensor performance including the limit of detection (LOD) and specificity. Researchers have introduced machine learning (ML) to bioreceptor-free biosensors to bridge this trade-off gap, improving the LOD and specificity [22]. In a sense, ML can be used to take the place of a bioreceptor by reintroducing specificity during data analysis. This is made possible by powerful ML techniques capable of detecting subtle patterns in sensor responses.

While this approach has demonstrated success, there are still several challenges that these systems must overcome. A major challenge being faced is model generalizability. Since many models rely on subtle patterns in the data, they can be quite sensitive to underlying data changes. This can make the models susceptible to error when faced with sensor drift or replacing parts of the system [14].

Since the scope of this review is quite large and covers all bioreceptor-free biosensors that utilize ML, there are a few points to clarify. Many subsets of our scope have received thorough attention and review. For instance, the use of ML for Enose and Etongue [23–27] and SERS-based

biosensors [28] have previously been described. Since the literature is rich in these areas, we realize that all recent original research cannot be adequately covered here. Rather, our intent is to provide a unified discussion of the relevant methods and challenges to give a bigger picture. We also would like to acknowledge that there is a complementary review in the literature discussing the use of ML in biosensing in general [29], but not for biosensors that are bioreceptor-free.

In this review, we will give the current state of using ML to enhance the performance of bioreceptor-free biosensors. Section 2 briefly introduces the types of biosensors that have most benefited from ML. Section 3 provides some background on machine learning algorithms and how their performance can be assessed. Section 4 covers electrochemical biosensors, with particular emphasis on Enose and Etongue. Successful methods are discussed as well as some of the challenges and how they are being addressed with ML. Section 5 discusses optical biosensors, notable for imaging- and SERS-based biosensors. Additional considerations and future perspectives are dis-cussed in Section 6 including what currently prevents many of these systems from being commercialized and what directions may be taken. We also present some considerations on best practices for ML in biosensing, especially regarding communication of methods and reproducibility.

2. How Biosensors Can Benefit from Machine Learning

Biosensors in the classic definition are sensors that utilize a bioreceptor such as antibody, enzyme, peptide, nucleic acid, etc. A bioreceptor binds to a target biological molecule and generates a signal when coupled with a transducer. Biosensors have evolved to a wide range of transducer types including electrochemical, optical, and spectroscopic biosensors. Traditionally, it is the bioreceptor that provides specificity and sensitivity to the biosensor. Increasingly, however, researchers are developing biosensors that lack a specific bioreceptor. A typical example is a semi-

specific chemical sensor array, termed Enose (from gas), or Etongue (in solution). Since such a sensor's specificity is not provided by the bioreceptor, a fingerprinting technique is used to recognize signal patterns indicative of a particular analyte. Frequently, machine learning techniques are employed to detect these patterns and provide specificity.

The use of machine learning to enhance the performance (e.g., specificity, sensitivity, and LOD) of bioreceptor-free biosensors is not limited to chemical sensor arrays. It has been employed in various biosensor mechanisms. Some of the most famous examples aside from Enose and Etongue are imaging-based biosensors and SERS-based biosensors. Additionally, the use of machine learning for biosensors is not limited to those that lack bioreceptors. Cui et al. [29] cover several examples of traditional biosensors employing machine learning to enhance performance.

Table A-1 provides an overview of the tasks for which machine learning has been applied, the specific algorithms used, and the relevant papers. More information on the algorithms themselves can be found in Section 3. Additionally, Table A-2 gives a comparison of each of the major biosensing mechanisms including data type and appropriate feature engineering and ML methods. All information in Table A-2 comes from Table A-1 and serves as a higher-level summary.

Table A - 1

Machine learning tasks and algorithms used in biosensing.

Biosensing Mechanism	Task	Target	Algorithm	Ref.
Electrochemical				
CV	Regression	Maleic hydrazide	ANN	[30]
CV	Classification	Industrial chemicals	LSTM, CNN	[31]
Enose	Feature extraction	Harmful gases	PCA	[32]
	Classification		DT, RF, SVM	
	Regression		SVR	
Enose	Regression	Formaldehyde	BPNN	[33]
Enose	Classification	Chinese wines	BPNN	[34]

	Target task change	Chinese liquors	Transfer learning	
Enose	Sensor drift compensation for classification	Gases	JDA	[35]
			DTBLS	[36]
			TrLightGBM	[37]
			ELM	[38]
Enose	Sensor drift compensation & noise reduction	Bacteria	ELM	[39]
EIS	Classification	Breast tissue	ELM + SVM	[40]
EIS	Classification	Milk adulteration	*k*-NN	[41]
EIS	Classification	Breast tissue	RBFN	[42]
EIS	Feature extraction	Avocado ripeness	PCA	[43]
	Classification		SVM	
EIS & EIT	Classification	Prostatic tissue	SVM	[44]
Etongue	Taste classification	Tea storage time	CNN	[45]
	Increase generalizability		Transfer learning	
Etongue	Feature Extraction	Beverages	*t*-SNE	[46]
	Classification		*k*-NN	
Etongue	Classification	Cava wine age	LDA	[47]
Etongue	Regression	Black tea theaflavin	Si-CARS-PLS	[48]
Optical				
Colorimetric	Classification	Plant disease VOCs (blight)	PCA	[49]
Diff. contrast microscopy	Digital staining & domain adaptation	Leukocytes	GAN	[50]
Fluorescence imaging	Classification	Microglia	ANN	[51]
FTIR imaging	Digital staining	H&E stain	Deep CNN	[52]
Lens-free imaging	Image reconstruction	Blood & tissue	CNN	[53,54]
		Herpes		[55]
Lens-free imaging	Image reconstruction & classification	Bioaerosol	CNN	[56]
Multi-modal multi-photon microscopy	Digital staining & modal mapping	Liver tissue	DNN	[57]
Multispectral imaging	Classification	Pollen species	CNN	[58]
Quantitative phase imaging	Digital staining	Skin, kidney & liver tissue	GAN	[59]
Raman spectroscopy	Feature extraction	Thyroid dysfunction biomarker	PCA	[60]
	Classification		SVM	
TLC-SERS	Feature extraction	Histamine	PCA	[61]
	Quantification		SVR	
SERS	Exploratory analysis	Malachite green & crystal violet	PCA	[37,62]
	Quantification		PLSR	

SERS	Quantification	Methotrexate	PLSR	[63]
SERS	Classification	Oil vs lysate spectra Leukemia cell lysate	*k*-means clustering	[64]
	Dimension reduction		PCA	
	Classification		SVM	
SERS	Dimension reduction	Levofloxacin	PCA	[38,65]
	Regression		PLSR	
SERS	Quantification	Potassium sorbate & sodium benzoate	PLSR	[66]
SERS	Dimension reduction & regression	Congo red	PCR	[39,67]
SERS	Dimension reduction	Mycobacteria	PCA	[40,68]
	Classification		LDA	
SERS	Quantification	Biofilm formation	PLSR	[41,69]
SERS	Feature extraction	Non-structural protein 1	PCA	[70,71]
	Classification		BPNN, ELM	
SERS	Exploratory analysis	Pollen species	PCA, HCA	[72]
	Classification		ANN	
SERS	Feature extraction	Human serum	KPCA	[73]
	Classification		SVM	

Note. CV = cyclic voltammetry; ANN = artificial neural network; LSTM = Long short-term memory; PCA = principal component analysis; DT = decision tree; RF = random forest; SVM = support vector machine; SVR = support vector regression; BPNN = back-propagation neural network; JDA = joint distribution adaptation; DTBLS = domain transfer broad learning system; GBM = gradient boost machine; ELM = extreme learning machine; EIS = electrical impedance spectroscopy; EIT = electrical impedance tomography; k-NN = k-nearest neighbor; RBFN = radial basis function network; CNN = convolutional neural network; t-SNE = t-distributed stochastic neighbor embedding; Si-CARS-PLS = synergy interval partial least square with competitive adaptive reweighted sampling; FTIR = Fourier transform infrared; VOC = volatile organic compound; GAN = generative adversarial network; DNN = deep neural network; TLC = thin layer chromatography; SERS = surface enhance Raman spectroscopy; PLSR = partial least squares regression; PCR = principal component regression; LDA = linear discriminant analysis; HCA = hierarchical cluster analysis; KPCA = kernel principal component analysis.

Table A - 2

Summary of data types and useful ML methods for biosensing mechanisms.

Biosensing Mechanism	Description of Data	Feature Extraction	ML Model
CV	Cyclic voltammogram		ANN, LSTM, CNN
EIS	Nyquist plot	PCA	*k*-NN, ELM, SVM, RBFN
Enose	Multivariate	PCA	DT, RF, ELM, SVM, BPNN
Etongue	Multivariate	PCA, *t*-SNE	LDA, *k*-NN, CNN, PLS

Lens-free imaging	Image		CNN
Digital staining	Image		Deep learning, GAN
SERS	Spectrum	PCA, KPCA	PLSR, LDA, SVM, SVR, BPNN, ELM

3. A Brief Tour of Machine Learning

In simple terms, machine learning aims to learn patterns in data to make predictions on new data. Generally, this prediction is either categorical classification (into one of a set of classes) or regression (continuous numerical output). In machine learning terms, the data used for prediction (i.e., biosensor data) are termed features or predictors. The set of features associated with one "observation" (e.g., biosensor data from one sample) is termed the feature vector.

3.1. Feature Engineering

Frequently, the predictor variables (feature vector) are not the raw biosensor data. One of the most challenging parts of using machine learning is the construction of the feature vector from the raw data. This process is termed feature engineering and mostly entails finding the relevant information from the data to aid the machine learning algorithm's performance. Common feature engineering steps include denoising, normalization, and rescaling.

One of the most powerful feature engineering processes is dimension reduction. This reduces a large number of features to a smaller number of features while minimizing information loss. Perhaps the most common method of dimension reduction is principal component analysis (PCA) [74], which reduces the original set of variables to a smaller set of independent variables termed principal components (PCs). The effectiveness of PCA to represent the data can be assessed by the amount of variance in the data explained by the PCs. Since PCA determines the PCs based on the eigenvectors' directions in the feature space, data must first be centered and rescaled to avoid bias toward those variables with a larger magnitude. Another common dimension reduction

algorithm is linear discriminant analysis (LDA), which also produces a smaller number of variables but is supervised and optimally maximizes class separation [75]. Other more complex dimension reduction methods exist including artificial neural networks (ANN), as discussed in Section 3.3. ANN is typically used as a supervised machine learning method, while it has occasionally also been used for dimension reduction.

3.2. Unsupervised Vs. Supervised

The two broad categories of machine learning algorithms are unsupervised and supervised [76]. In unsupervised methods, data labels are not provided during model training, while in supervised methods, they are. An example of an unsupervised algorithm is cluster analysis, used to group similar data. Unsupervised methods are less common in biosensing since we generally know what kind of prediction(s) we would like the model to make. A notable exception is PCA, as mentioned in Section 3.1. While PCA may be considered an unsupervised machine learning method, its use has recently been limited to dimension reduction (one of feature engineering processes) prior to supervised machine learning analyses.

3.3. Classification Algorithms

Among the supervised methods, classification algorithms are some of the most well-known. Classification gives prediction in the form of a class label (e.g., which bacteria species is present), thus, the output is inherently categorical. Briefly, some of the most common classification algorithms are presented in the following.

k-nearest neighbors (*k*-NN): One of the simplest classification algorithms, *k*-NN is a distance-based classifier. Class is predicted as the most common class of the *k*-nearest neighbors in the feature space [77]. In the example shown in Figure A-1, the feature space is two dimensional (with variables x1 and x2) and the value of *k* is 4. In k-NN, the number of neighbors used for

assignment, *k*, is a hyperparameter (i.e., a model parameter that is not optimized during the training process itself). As with most ML models, hyperparameter selection may strongly influence performance [78].

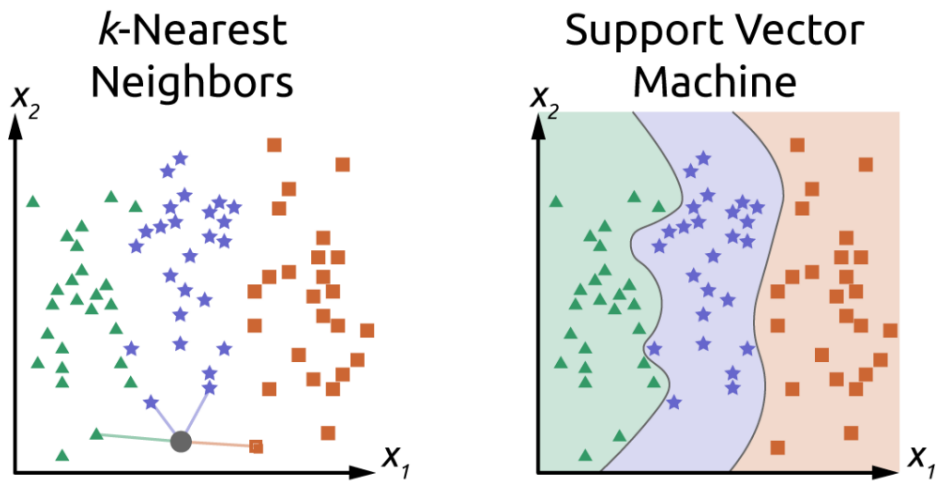

Figure A - 1

Comparison of classification technique using k-NN and SVM. In k-NN, four nearest neighbors are shown contributing to the gray point's assignment. Classification of the gray point is the blue star class. In hypothetical SVM with nonlinear kernel, new data are classified in which region the point lies. In both examples, the feature space consists of two dimensions. Classification could be, for example, bacterial species like *E. coli*, *Salmonella spp.*, *Pseudomonas spp.*, *Staphylococcus spp.*, *Enterococcus spp.*, etc. In practical applications, the feature space has many more dimensions, where decision boundaries for SVM are hyperplanes in the $(n-1)$ dimension for an n-dimensional feature vector.

Support vector machine (*SVM*) is a non-probabilistic, binary, linear classifier [79]. SVM relies on the construction of hyper-plane boundaries in the feature space to separate data of different classes. Although SVM itself only accounts for linear separation of classes (i.e., hyper-plane boundaries must be "flat"), the data may be mapped to a higher-dimensional feature-space

using the "kernel trick" [80]. Some of the most common kernels are radial basis function and Gaussian. When the hyperplane boundaries are projected back into the original feature space, they allow for non-linear boundaries, as shown in Figure A-1. Additionally, there are methods allowing SVM to be used for multi-class prediction [81]. The placement of hyperplanes is determined by minimizing the distance between the hyperplane and several of the points closest to the boundary between classes. SVM's robustness against outliers is improved by a soft margin. This allows for a certain quantity of misclassifications, which are presumably outliers, to improve the separation of the other observations [82]. While SVM shows resilience against outliers and performs well in high-dimension feature spaces, it is prone to over-fitting, especially when using non-linear kernels [83]. Overfitting is when the model performs well on training data but performs poorly when generalized to unseen data.

Linear discriminant analysis (*LDA*): In addition to dimension reduction, LDA can be used for classification. Other related algorithms allow for non-linear classification such as quadratic discriminant analysis (QDA) [84]. One of the limitations of LDA and its relatives is that they assume the data are normally distributed.

Decision tree (*DT*) and *random forest* (*RF*): In tree-based models such as decision tree (DT), the feature vector starts at the tree's "trunk," and at each branching point a decision is made based on the learned decision rules. The end classification would then be at the terminal or "leaf" node that the instance results. DTs can be used for classification and regression [85]. When the target variable is categorical, it is referred to as a classification tree; when the target variable is numerical and continuous, it is referred to as a regression tree [86]. Random forest (RF) is so called because it can be considered a forest of decision trees (Figure A-2) [87]. There are many RF

architectures, but in all instances, the classification from each decision tree contributes to the overall classification for an observation.

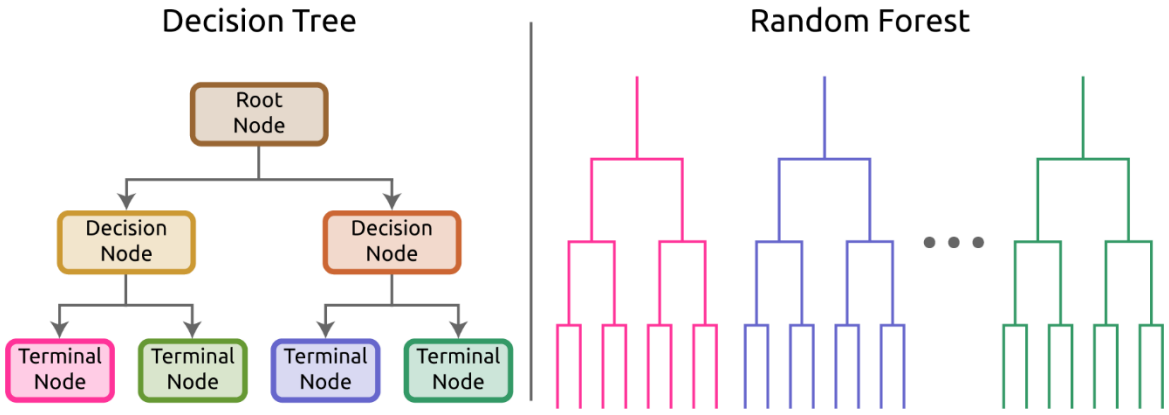

Figure A - 2

Decision tree (DT) showing nodes at which binary decisions are made on features. Terminal node dictates model prediction. Actual DTs have many more nodes than shown here. Random forest (RF) shown as a series of distinct decision trees.

Artificial neural network (ANN) draws inspiration from biological neural networks (i.e., neurons in the brain) and is composed of a collection of connected nodes called artificial neurons (see Figure A-3). ANNs can be used for classification and regression. As mentioned earlier, ANN can be used for dimension reduction prior to supervised machine learning. There are a large variety of ANN structures such as (1) recurrent neural network (RNN) [88], (2) extreme learning machine (ELM) [89], and (3) deep learning algorithms such as the convolutional neural network (CNN) [90], deep belief network [91], and back-propagation neural network (BPNN) [92]. "Deep" indicates several hidden layers. ANN architectures have many hyperparameters such as the number of hidden layers, connectedness, and activation functions [93].

One of the aspects that makes ANN so powerful is that features do not need to be well-defined real numbers. This allows them to excel at working with data such as images for which

extracting numerical features would be difficult and inefficient. One limitation of ANNs is that they require a large amount of data for effective training. In some settings, training data sparsity can be mitigated through a generative adversarial network (GAN) using back propagation [94].

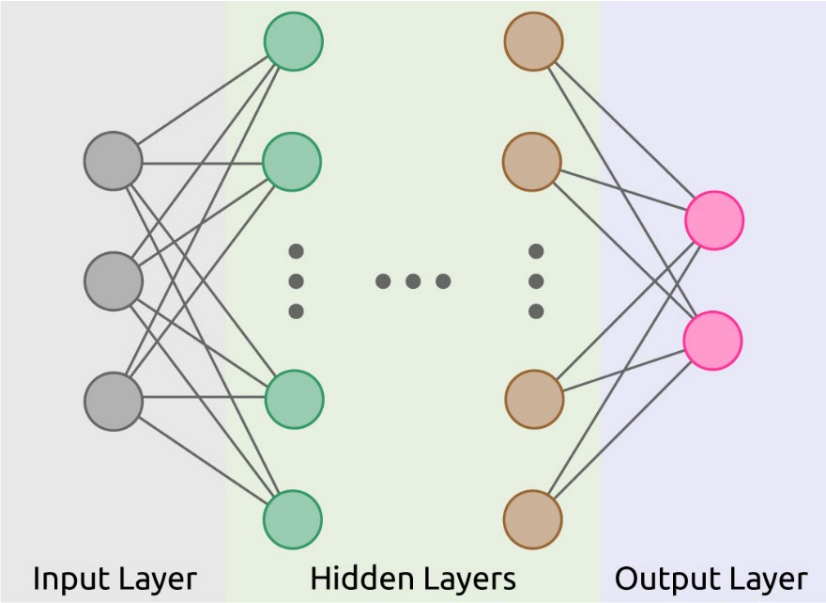

Figure A - 3

Artificial neural network (ANN) showing nodes of the input, hidden, and output layers.

Common classification model performance metrics are accuracy, precision, sensitivity (also known as recall), specificity, and $F1$. For binary classification with labels "positive" and "negative", they are defined as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$precision = \frac{TP}{TP + FP} \tag{7}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$specificity = \frac{TN}{TN + FP} \tag{9}$$

$$F1 = \frac{2 \times precision \times sensitivity}{precision + sensitivity} \quad (10)$$

where *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative.

3.4. Regression Algorithms

In contrast to classification, the prediction made by a regression algorithm is a numeric value from a continuous scale (e.g., glucose concentration in blood). A simple regression example fits a linear model of the form y = mx + b, where a model is built for the prediction of the output variable y based on the input variable x, and the coefficients m and b are "learned" from the data. The learning is typically done by the least-squares regression approach, minimizing the sum of the squared residuals. The following are some of the most common regression algorithms.

Multilinear regression (*MLR*) is a simple regression model, which expands the above linear model example, accounting for multiple input variables. This model shows how it can be difficult to determine when an algorithm becomes sophisticated enough to be considered "machine learning".

Support vector regression (*SVR*) is an adaptation of SVM used for regression problems. Like SVM, SVR can utilize kernels to allow for non-linear regression. An advantage of SVR over traditional regression is that one need not assume a model that might not be accurate. For instance, with linear regression, there is an assumption that the data distribution is linear. SVR does not require such pre-determined assumptions [95].

Regression tree is an adaptation of DT for regression. Regression tree has the advantage that it is non-parametric, implying that no assumptions are made about the underlying distribution of values of the predictors [86].

Artificial neural network (*ANN*) is also widely used for regression problems, and many varieties exist, some of which were mentioned previously.

A large variety of metrics exist for regression model performance. Since there are too many to define here, for further reading, we suggest the study by Hoffman et al. [96] to learn more. Some of the most common metrics are briefly presented here. Root mean squared error (RMSE) and mean absolute error (MAE) have the benefit that their units are the same as the output (predicted) variables, but this makes the metrics less universally understandable. Normalized root mean squared error (NRMSE) partially resolves that. Coefficient of determination, R2, on the other hand, is unitless and $R^2 \leq 1$, where a value near 1 is generally considered good performance (although this is a bit oversimplified).

3.5. Model Performance Assessment

Frequently, researchers will try various models and compare their performance. The value of the performance metrics listed above can be treated as random variables and statistical analyses can be used to test hypotheses regarding which model is better [96]. While this sounds simple, it can be nuanced: for instance, when working with a classification model, which metric is most important for your application? In some cases, specificity may be more important than accuracy, for instance. Additionally, when using statistical tests to compare model performances, certain assumptions are made, and their validity should be assessed such as when using NRMSE, as it is assumed that noise affecting the output is random and normally distributed.

The best practice for model selection, tuning, and performance assessment is to split the data into 3 sets: training, testing, and validation. For example, if the database consists of 1000 observations, 100 (10%) are assigned to the validation set and the remaining 900 (90%) are split between the training and test sets as 810 (90%) for training, 90 (10%) for testing. The model is then trained on the labeled training set. Model selection and hyperparameter tuning is conducted based on model performance when challenged using the test set. In addition to train–test splitting,

cross-fold validation can be used on the training set when tuning hyperparameters or comparing models [97]. Train–test splitting and cross-validation are most important when you intend to generalize the model to predict new, unseen data [96]. Final model performance validation is conducted on the validation set, which should not be used until all model selection and hyperparameter tuning have been completed.

4. Electrochemical Bioreceptor-Free Biosensors

Since their inception, electrochemical biosensors have become extremely popular. In traditional electrochemical biosensors, the bioreceptor interacts with the target to generate a signal at the electrical interface. A widespread scheme is an enzyme (*e.g.,* glucose dehydrogenase or glucose oxidase) interacting directly with the target analyte (*e.g.,* glucose), catalyzing a redox reaction that generates a signal at the electrical interface [98]. Electrical interfaces include metal electrodes, nanoparticles, nanowires, and field-effect transistors (FET) [99].

It is also possible to eliminate the biorecognition element (=bioreceptor, e.g., an enzyme) in electrochemical biosensors. Voltametric sensors described in Section 4.1 can detect biomolecules based on direct interaction with the electrical interface [30]. Electrical impedance spectroscopic biosensors can also detect subtle differences in a solution or material's electrical impedance, as discussed in Section 4.2. Alternatively, we can use an array of chemical or physical sensors varying the electrical interface to create multi-dimensional data. Machine learning-based pattern recognition is used to identify the target analyte. Two of the most common sensor arrays are termed Enose and Etongue, which are covered in Section 4.3.

4.1. Cyclic Voltammetry (CV)

Voltammetry sensors apply electric potential to a "working" electrode and measure the current response, which is affected by analyte oxidation or reduction [100]. Cyclic voltammetry

(CV) is a specific voltammetry technique in which the potential is swept across a range of values, and current response is recorded. These CV curves (cyclic voltammograms) can serve as a fingerprint of the sensor response. A typical CV curve is shown in Figure A-4A.

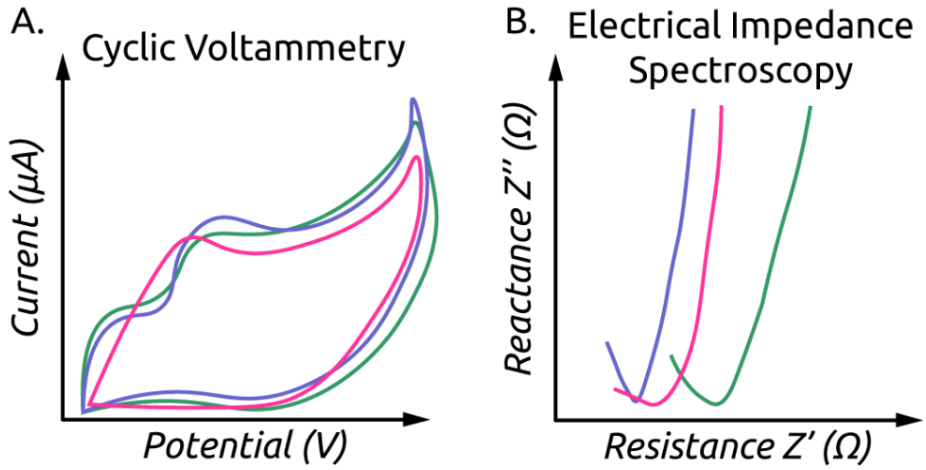

Figure A - 4

(**A**) Hypothetical cyclic voltammograms for three samples. (**B**) Hypothetical Nyquist plot obtained through EIS showing curves for three samples.

CV biosensors often employ bioreceptors to provide specificity in the interaction between target analyte and electrode surface. However, there has also been research on utilizing more complex electrode surface structures and modifications to allow for semi-specific interaction with the target analyte without the need of a bioreceptor. Sheng et al. [30] describe a compound electrode utilizing Cu/PEDOT-4-COOH particles for CV detection of the phytoinhibitor maleic hydrazide. They found that several regression models had poor performance for modeling the sensor current response with respect to target concentration. However, they employed an ANN with great success for the same regression task. The result is that their detection range is broader than comparable methods by an order of magnitude at each extreme (detection range = 0.06–1000 μM and LOD = 0.01 μM).

4.2. Electrical Impedance Spectroscopy (EIS)

Electrical or electrochemical impedance spectroscopy (EIS) is an analytical technique that provides a fingerprint of the electrical properties of a material. EIS is performed by applying a sinusoidal electric potential to a test sample and recording the impedance (both resistance and reactance expressed in a complex number) over a range of frequencies [101]. Frequently, an equivalent circuit model is fitted to EIS data to provide a fingerprint of the material properties [101]. Figure A-5 shows an equivalent circuit diagram for EIS being performed on a single cell suspension. An example EIS spectrum is shown in Figure A-4B. It is the classification and regression on such fingerprints that machine learning tends to be well suited.

A simple example of this is the use of k-NN on EIS data for the detection of adulteration in milk [41]. In this work, the feature space was composed of resistance at a certain temperature and pH. They demonstrated good accuracy of 94.9%. However, the data were highly imbalanced, and in the example classification plot [41], one of the three unadulterated samples were misclassified, a 66% specificity.

More robust classification has been performed using SVM. One example is for the assessment of avocado ripeness [43]. This work describes using PCA for feature extraction, resulting in two PCs that explain >99.3% of the variance. SVM is then used for classification based on the first two PCs. SVM for EIS was also described by Murphy et al. [44] for classification of malignant and benign prostatic tissue. However, instead of using PCA for feature extraction, equivalent electrical circuit model parameters were used as predictors. The feature vector size was 2160, consisting of four electrical features for each of eleven frequencies across multiple electrode configurations. Classification was also performed on electrical impedance tomography (EIT) data from the same samples using SVM. Both showed good classification performance, though the

authors mention that EIT may be preferable since the measurements are not dependent on probe electrical properties, and thus can be compared more easily to other studies.

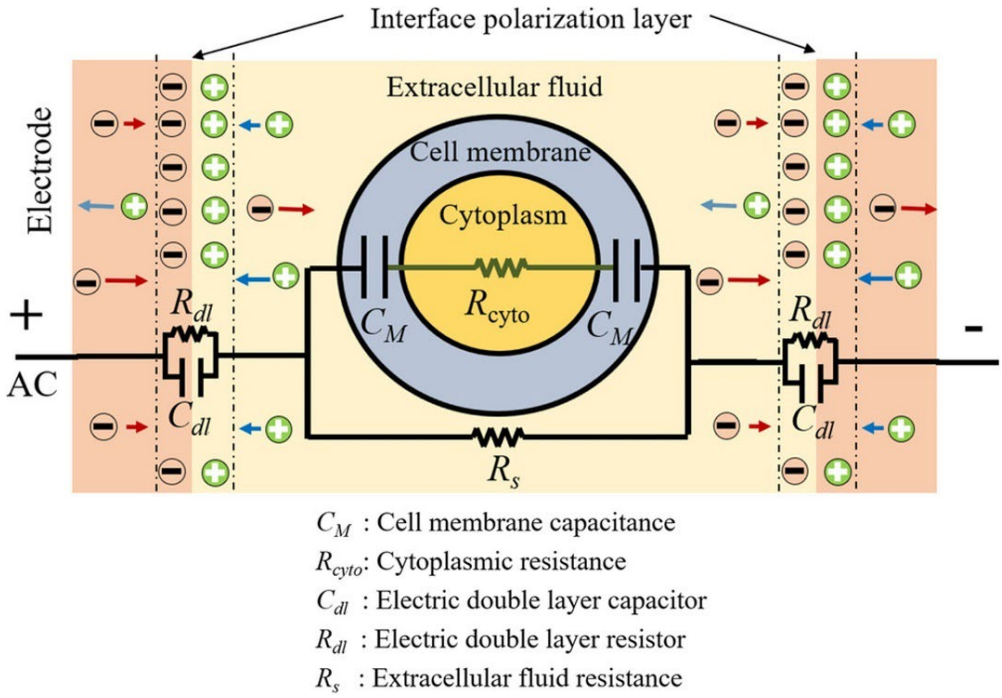

Figure A - 5

Equivalent circuit diagram of single cell suspension. Reproduced with permission from [102] without modification. Copyright 2020 John Wiley and Sons.

While SVM is renowned for its tolerance of outliers, this is a trade-off in that data points not near the boundary between classes do not contribute to defining class attributes. However, ANNs preserve more of this information for prediction. When the number of observations or predictors are small, this can lead to overfitting. However, with sufficient data size, ANNs can preserve predictive information and be robust against outliers and overfitting. These attributes have been utilized for EIS based classification of breast tissue [40,42]. Both works use the same publicly available dataset of EIS measurements from freshly excised breast tissue [103], made available on the University of California, Irvine (UCI) Machine Learning Repository [104]. The dataset

contains nine spectral features from EIS. Daliri [40] describes using three ELMs, each with different numbers of nodes, and feeding the output of the three ELMs (extreme learning machines) into SVM for classification. This method showed improved performance over previous methods for the same dataset such as LDA [105]. Helwan et al. [42] compared both BPNN and radial basis function network (RBFN) for the same task. Both methods showed an improvement over ELM-SVM as described by Daliri [40], with RBFN performing better than the BPNN including improved generalizability (i.e., classification performance on new data).

It is seen that in the case of EIS classification, node-based models have shown improved performance over other models. This can be seen most clearly when comparing classification accuracy for those methods that utilized the same dataset. The RBFN and BPNN had the highest classification accuracy, with 93.39% and 94.33%, respectively [42]. The next best performance was achieved by the ELM-SVM, achieving 88.95% accuracy [40]. These results show marked performance increase over LDA [105]. Model performance is greatest in those models that do not utilize distance for classification (i.e., SVM and LDA). While distance-based classifiers are robust to outliers, in these EIS datasets, performance benefitted by node-based classification.

4.3. Enose and Etongue

Enose and Etongue are named in analogy to their respective animal organs. Both sensor types rely on an array of semi-specific sensors, each of which interacts to a different degree with a wide range of analytes. Figure A-6 shows a comparison between Enose and Etongue alongside the analogy to their respective biological systems [27,106]. The sensor arrays can be composed of any variety of sensors. The following chemical gas sensors have been used in Enose systems: metal oxide (MOX) gas sensor, surface or bulk acoustic wave (SAW and BAW) sensors, piezoelectric sensor, metal oxide semiconductor field-effect transistor (MOSFET) sensor, and conducting

polymer (CP) based sensor [107]. Similarly, a variety of sensors can be employed in Etongue systems such as ion-selective field-effect transistor (ISFET) and light-addressable potentiometric sensor (LAPS) [108].

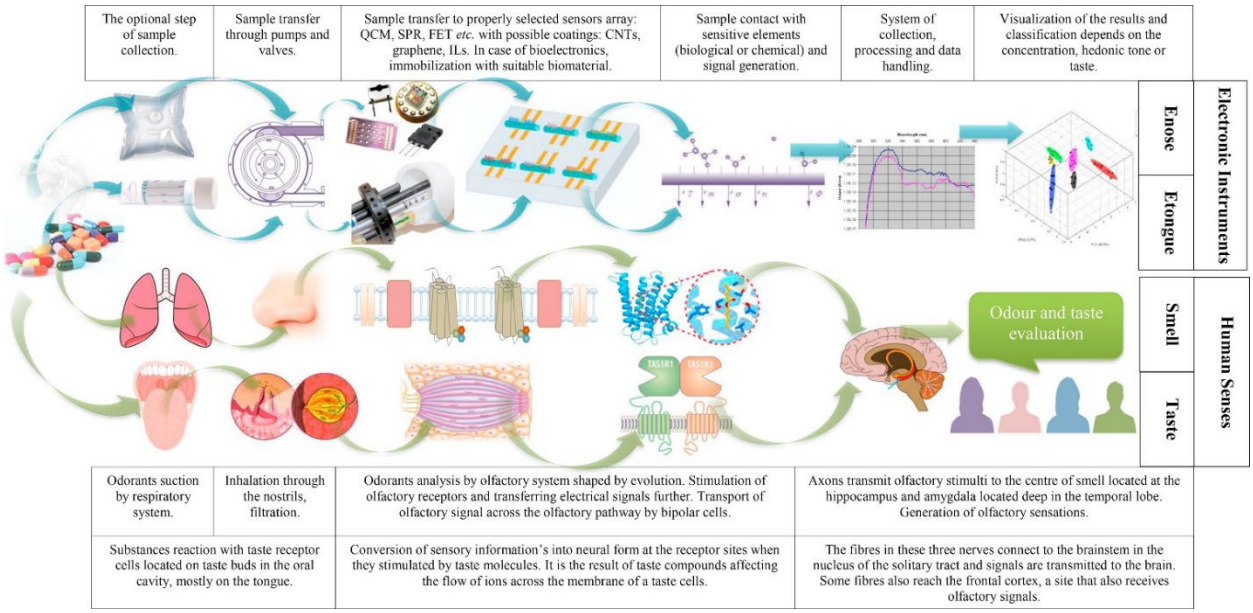

Figure A - 6

Comparison of operation principle of Enose and Etongue, and the analogy to the biological systems. Reproduced with permission from [27] without modification. Copyright 2019 Elsevier.

Analyte presence, or a more general attribute such as odor or taste, is detected through pattern recognition of the sensor array response. For pattern recognition on this naturally high-dimensional data, machine learning techniques are an obvious choice. Scott et al. provided a relevant and succinct paper on data analysis for Enose systems [23]. As discussed in Section 3 of this review, feature engineering is critical in any machine learning pipeline. Yan et al. [24] provide a review article on the feature extraction methods for Enose data. For non-linear feature extraction of Etongue data, Leon-Medina et al. [46] give a great comparison of seven manifold learning methods.

A vast number of papers exist detailing such systems and their use of machine learning. As such, it would be infeasible to cover all of them adequately. For this review, a higher-level analysis is presented by looking at the conclusions reached in the review papers covering this topic as well as a few notable examples of specific papers. Of particular interest is which algorithms had the most success with Enose and Etongue sensors or applications.

A common task of Enose is the prediction of "scent", which is a classification problem. Before the application of the classification algorithm, it is common to perform dimension reduction. PCA is the most common choice for this task, although independent component analysis (ICA, a generalization of PCA) has shown success [25]. PCA has been shown to improve the performance over classification algorithms alone for the piezoelectric Enose [25]. The two classifiers most commonly in use are SVM [109,110] and various ANN methods [25,111]. In addition to classification problems, Enose may be used for analyte concentration prediction. One example is the use of MOS (metal oxide semiconductor) gas sensors for formaldehyde concentration assessment. In this case, the back-propagated neural network (BPNN) outperformed radial basis function network (RBFN) and SVR [33]. In another instance, with the single nickel oxide (NiO) gas sensor, PCA with SVR was utilized for harmful gas classification and quantification [32]. In cases where the amount of data is not large, SVM may be advantageous over node-based models (ANNs) for its resilience against outliers and overfitting.

While Enose and Etongue systems have shown great promise for non-destructive analytical devices, there are challenges that have limited their use in commercial settings. Several challenges involve changes in the sensor data, which affect the performance of the trained model. A common phenomenon is when the sensor array response changes over time or upon prolonged expose under identical conditions. Such change in sensor response is referred to as sensor drift and can greatly

affect the trained models' performance [14]. Another way in which the sensor response may change is if a sensor in the array becomes defective and must be replaced, as it is difficult to replace it with one that responds identically, largely due to variability in manufacturing [112,113]. For both challenges, time consuming and computationally expensive recalibration may be necessary.

The issue of needing retraining due to underlying data distribution changes is commonly addressed through transfer learning in many machine learning settings. Transfer learning is a computational method for minimizing the need for retraining when either the data distributions change (e.g., sensor array response to an analyte) or the task changes (e.g., new classes of analytes are being detected).

Transfer learning has been extensively employed to counter Enose sensor drift and reduce the need for complete retraining [35–38]. It has also been used to reduce the deleterious effect of background interference [39,114]. Although several of the above papers [35,36,38,39] demonstrate the efficacy of their approach on a shared sensor drift dataset shown in Figure A-7 [115], ranking of the methods is difficult due to inconsistent benchmarking metrics. As mentioned previously, the data distribution may also change due to replacing a sensor with a new sensor, or when attempting to apply a trained model to a theoretically identical array with differences due to manufacturing variability. Transfer learning, specifically using ANN, has demonstrated decent recalibration [116].

One instance of utilizing transfer learning for target task change was demonstrated by Yang et al. by training an Enose classifier on wines (source task) and applying it to classify Chinese liquors (target task) while only retraining the output layer [34]. Interestingly, transfer learning has been used much less commonly for Etongue systems, although they also face sensor drift. However, Yang et al. utilized transfer learning to improve the generalizability of their Etongue

[45]. In this work, they demonstrate the superiority of their transfer learning trained CNN over other methods such as BPNN, ELM, and SVM for tea age classification.

A trend that has been gaining traction is data fusion to combine Enose and Etongue systems. The value of this can again be appreciated in how closely the senses of smell and taste are linked in animals [117], complementing each other to provide the most accurate assessment. Similarly, by using information from both Enose and Etongue, better analysis can be conducted. As illustrated in Figure A-8, data fusion can be performed at three levels: low, mid, and high [118]. Recently, mid-level fusion schemes have shown promising results for fusion of Enose and Etongue data [119,120], especially when performing PCA on the two systems and using those features for fusion before model training [121–123]. Such systems have also benefitted from the inclusion of a computer vision system in data fusion [121,124].

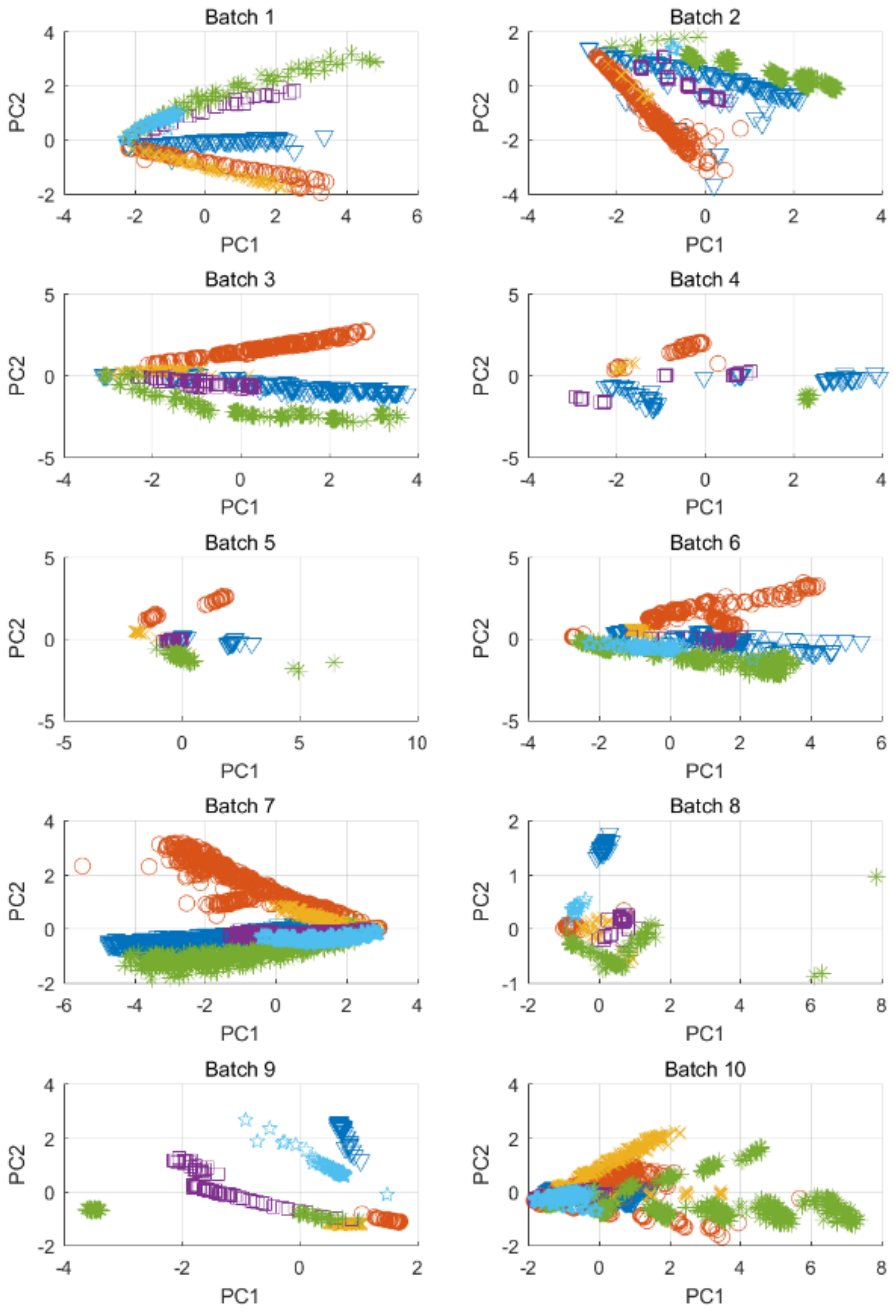

Figure A - 7

Gas sensor drift dataset from [36]. Each color represents a different gas. Each panel represents a measurement "batch" at various times spanning 36 months. Reproduced from [36] without modification, under Creative Commons Attribution 4.0 License.

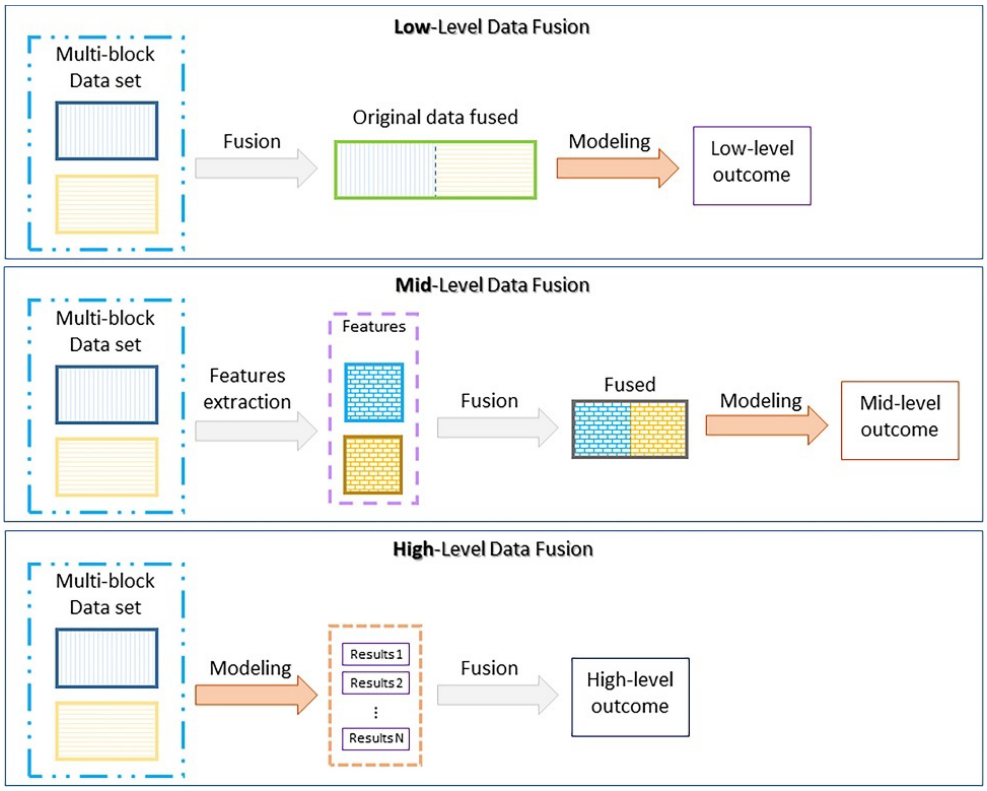

Figure A - 8

General scheme depicting the main differences among low-, mid-, and high-level data fusion. Reproduced with permission from [118] without modification. Copyright 2019 Elsevier.

Currently, another class of systems exist with the same goals as Enose and Etongue that utilize biochemical recognition elements, termed bioelectronic nose (bEnose) and tongue (bEtongue). These devices utilize biological elements such as taste receptors, cells, or even tissues for sensing [106,125]. These systems show impressive selectivity and sensitivity, especially when coupled with nanomaterials to aid in signal transduction from the biochemical recognition element [106,126]. Their major challenges, as with most biosensors, is stability and reproducibility of the biological element [106]. For these reasons, Enose and Etongue remain popular for their sensor stability. Continued efforts are necessary to improve sensitivity closer to their bioelectronic counterparts, especially regarding sensor design and feature extraction methods.

With such a large variety of sensors in use for Enose and Etongue systems, data processing can vary significantly. Of particular interest is finding appropriate feature extraction methods [23,24]. A huge variety of machine learning classification and regression methods have been employed, both on unsupervised dimensionally reduced feature vectors and classically extracted features. Transfer learning methods have been successful in allowing target task change with minimal retraining, especially when using node-based models. However, the challenges posed by sensor drift and manufacturing variability are still significant and will likely remain a focus for researchers over the next several years.

4.4. Summary of Electrochemical Bioreceptor-Free Biosensing

Many electrochemical bioreceptor-free biosensors employ chemical or physical sensor arrays coupled with machine learning. These are most obvious in Enose and Etongue systems, inspired by nature (humans and animals). Other systems generate multivariate spectral data also coupled with machine learning. In both cases, machine learning models can aid in analyte classification or quantification. Especially when using distance-based models, choice of feature extraction method is important to optimally capture the features relevant to the task (i.e., classification or regression). Node-based models, primarily ANNs often require less feature extraction pre-processing as this step is built into the model learning. Additionally, node-based models offer a great solution to target task change and noise elimination through transfer learning, often aided by integration through the back-propagation step so that only the final layer needs to be refined [34].

5. Optical Bioreceptor-Free Biosensors

The mechanisms of optical detection in biosensing are diverse. A classic example is the colorimetric lateral flow assay [127–129]. Mechanisms beyond colorimetry include fluorescence [130–132], luminescence [133], surface plasmon resonance [134], and light scattering [135,136].

Machine learning has been widely employed in optical biosensors. An example with similarities to Enose and Etongue is the bacterial bioreporter panel. Each bacterial bioreporter responds to target analytes in a semi-specific manner. Machine learning is used to discover patterns in the bioreporter panel response and relate them to analyte presence or concentration [137,138]. However, this review's focus is to discuss cases in which the bioreceptor is absent, so such sensors are not covered in detail.

Another prevalent use of machine learning for analyzing images as biosensor data is for image processing, especially segmentation [139–142]. The literature is rich in reviews on machine learning for image segmentation, and this technology is in no way specific to biosensors, so this review will not discuss those examples. However, the topic is essential to many biosensors, so it must be mentioned.

5.1. Imaging

Imaging sensors utilize an array of optical sensors such as a CMOS array (complementary metal-oxide-semiconductor array; the most used image sensor for digital cameras). Images of the specimen can be used to identify the target presence and concentration as the molecules exhibit different coloration, fluorescence, or light scattering, with varying morphology and spatial distribution. In this manner, several imaging biosensors have been developed to eliminate the need for labels and bioreceptors.

A growing field of imaging-based biosensors utilizes lens-free imaging techniques [143,144]. Since the images from lens-free imaging are not in focus, computational techniques are needed for image reconstruction, the most common of which is deep learning (mostly based on ANN with “deeper” layers) [53,54,145]. Lens-free imaging may be used to detect the aggregation of particles caused by bioreceptor–analyte interaction [55] (Figure A-9). However, an exciting application is the direct, label-free classification of particles by lensless holography. Wu et al. [56] presented a lensless holography biosensor for classifying pollen and spores. As with many of these systems, a CNN was used for image reconstruction. In this work, another CNN was used to classify the particles, yielding >94% accuracy.

Another work on the imaging classification of pollen utilizes multispectral imaging [58]. Again, a CNN was trained for classification, and a species-averaged accuracy of 96% was achieved for 35 plant species.

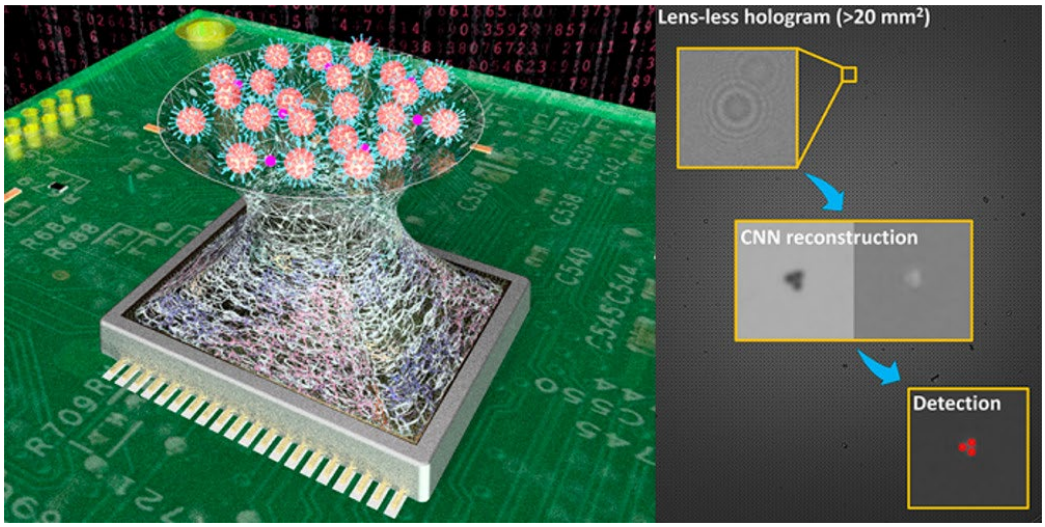

Figure A - 9

Lensless hologram reconstruction via CNN for particle aggregation detection. Reprinted with permission from [55] without modification. Copyright 2019 American Chemical Society.

Artificial neural networks (ANNs) have also found great success in the developing field of digital staining. Hematoxylin and eosin (H&E) stain is the most common stain for histology [146]. However, the quality of tissue staining is subject to many factors that can affect the diagnosis. Digital staining is an alternative in which tissue sections are imaged unstained, and a trained model generates an image simulating stained tissue (Figure A-10). Deep learning has been applied for digital staining on images acquired from a variety of methods including quantitative phase imaging [59], Fourier transformed infrared spectroscopy (FTIR) [52], and multi-modal multi-photon microscopy [57]. To overcome the issue of data scarcity and overfitting, researchers have frequently employed generative adversarial neural network (GAN) for medical imaging [147], which has shown promising results for digital staining model training [148]. Additionally, transfer learning has improved the model's generalizability to multiple domains [50].

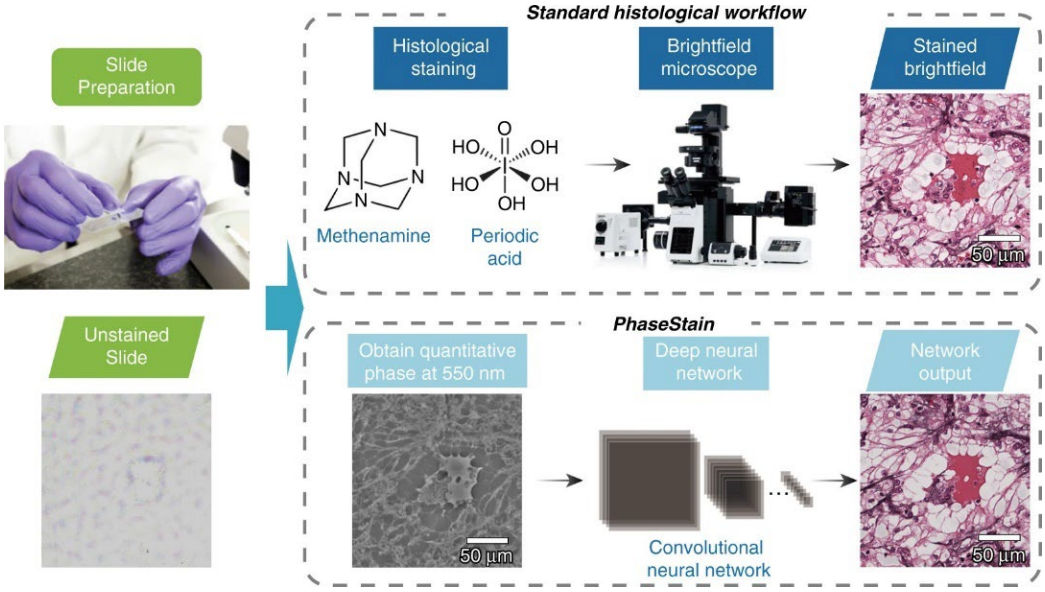

Figure A - 10

A quantitative phase image of a label-free specimen is virtually stained by a deep neural network, bypassing the standard histological staining procedure that is used as part of clinical pathology. Reproduced from [59] without modification under Creative Commons Attribution 4.0 License.

Fluorescence-based imaging biosensors are also worthy of mention. Sagar et al. [51] presented a microglia classification based on fluorescence lifetime utilizing ANN.

The applications of imaging biosensors are extensive. Indeed, the scope is too large to analyze all papers in this review. However, of particular importance to imaging biosensors is the ANN, especially the CNN. This preference is expected since CNN has shown exceedingly good performance in a variety of image classification contexts [149,150].

5.2. Colorimetry

One class of optical biosensors is the colorimetric biosensor. Currently, the applications of machine learning to enhance the performance of bioreceptor-free colorimetric biosensors are limited. This limitation is because the colorimetric biosensors (most notably lateral flow assays) mostly utilize bioreceptors (e.g., antibodies, enzymes, and aptamers) [98]. One example of such a bioreceptor-free biosensor is non-invasive plant disease diagnosis by Li et al. [49]. They utilized an array of plasmonic nanocolorants and chemo-responsive organic dyes that interact with volatile compounds from the plant. Their technique is similar to Enose and Etongue since it is a fingerprinting approach to the array response for classification. They used PCA, but do not cite an actual classifier, although they give performance metrics such as accuracy. At this time, it is unclear how the classification was performed on the PCA-transformed data.

Most colorimetric biosensors do not require machine learning due to their simplicity for readout. However, the arrays of bioreceptor-free (semi-specific) colorimetric sensors require machine learning-based classification in a way similar to Enose and Etongue. In these instances, they will likely benefit from the same treatment, namely dimension reduction by PCA and SVM classification.

5.3. Spectroscopy

Of the spectroscopic biosensing techniques, surface-enhanced Raman spectroscopy (SERS) has shown great success [151,152]. SERS is a vibrational surface sensing technique that enhances Raman scattering based on surface characteristics. Briefly, SERS utilizes incident laser light to induce inelastic scattering (Raman scattering) from the target analyte. The intensity of the Raman scattering is enhanced by interaction with the conduction electrons of metal nanostructures (SERS substrate). The enhancement of the Raman scattering is what makes SERS so sensitive. Researchers have reported enhancement factors of up to ten or eleven orders of magnitude [153]. Figure A-11 illustrates a SERS sensor for the analysis of breath volatile organic compound (VOC) biomarkers [154]. Due to the complex nature of the obtained spectral signal, various machine learning algorithms have been used to process SERS data in multiple contexts [28].

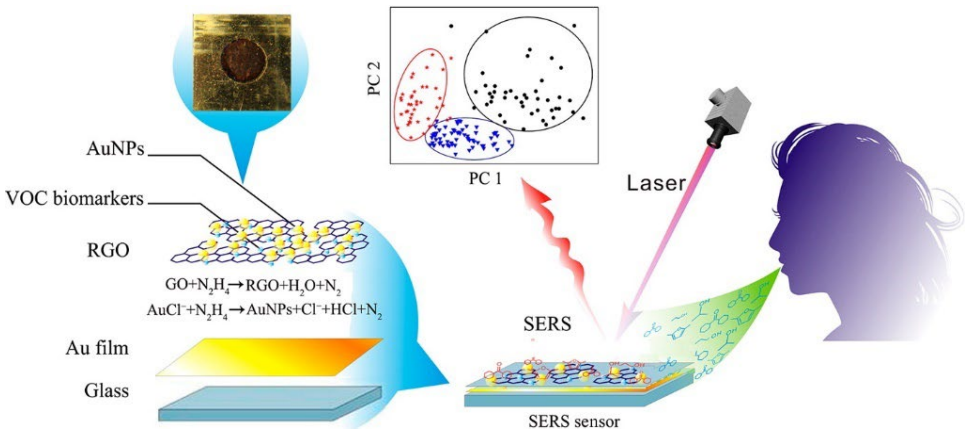

Figure A - 11

SERS sensor for analysis of breath VOC biomarkers utilizing AuNPs. Reprinted with permission from [154] without modification. Copyright 2016 American Chemical Society.

Although bioreceptors may be used to allow for specific binding of the target analyte to the SERS sensing surface [155,156], direct detection is also possible. Robust classification and regression algorithms can bring specificity and sensitivity to these biosensors. A simple yet

effective method for SERS based quantification is partial least squares regression (PLSR). PLSR has been used for a variety of quantification applications such as biofilm formation monitoring [69], blood serum methotrexate concentration [63], aquaculture toxins [62], and food antiseptics [66]. PLSR has the advantage of model simplicity with well-defined parameters, but it may be insufficient in modeling data with significant sources of noise.

Since the spectra have high dimensionality, dimension reduction is a frequent preprocessing step (Figure A-12). PCA is again popularly used as a dimension reduction or feature extraction step [60,61,64,65,68,70,71,73], or for exploratory analysis [62,72,157]. Once the spectra are remapped using PCA, a classifier or regression model is employed such as an extreme learning machine (ELM) [71], LDA [68], SVM [60,64,73], PLSR [65], or ANN [70]. An alternative to dimension reduction is utilizing the high dimensionality spectral data directly with a node-based algorithm such as ANN [72,158,159] and CNN [160,161].

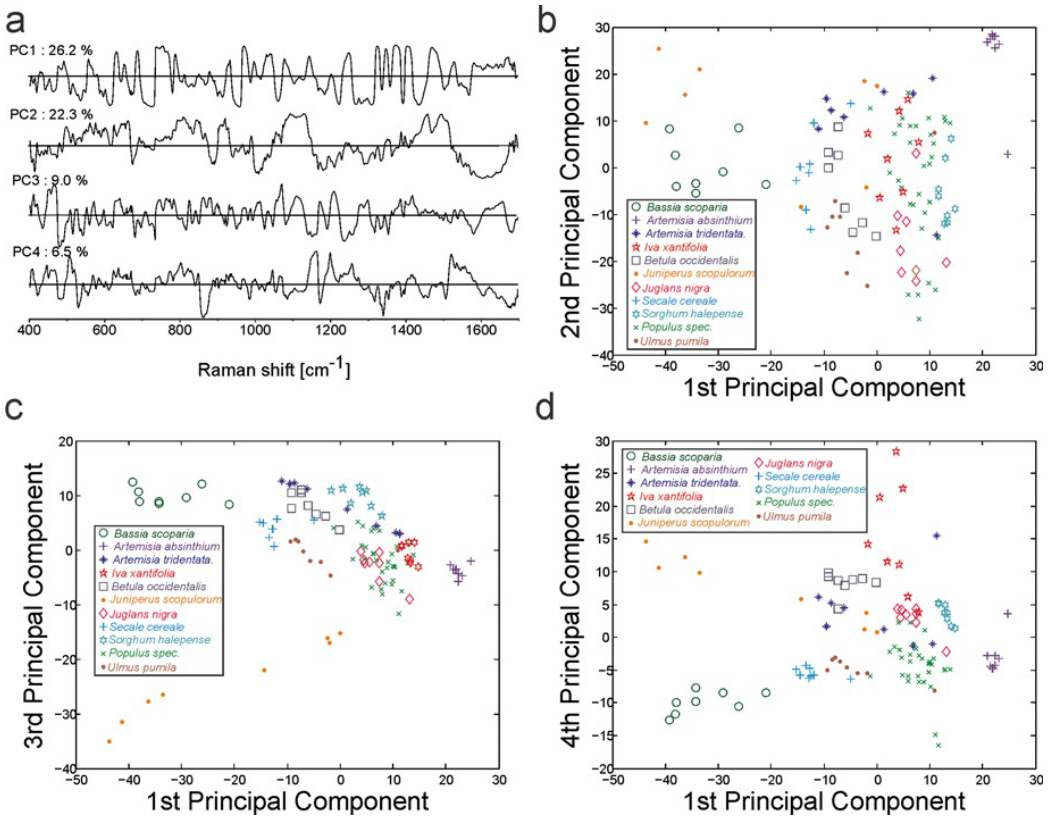

Figure A - 12

PCA results using the spectral range of 400–1700 cm-1 of 112 average SERS spectra from 14 different commercially available pollen species. Loadings of the first four PCs (a) as well as the scores of the first and second (b), first and third (c), and first and fourth PC (d) are shown. PCA was done with standardized first derivatives of the mean spectra of 500 vector-normalized spectra. Reprinted with permission from [72] without modification. Copyright 2016 John Wiley and Sons.

The reusability and generalizability of the trained models are often limited. Spectral response is affected not just by analyte presence but surface structure. Therefore, for the model to be reused on a new SERS biosensing dataset, the surface characteristics must be very similar. In terms of transfer learning, this is an issue of changes in the underlying data distributions. However, if the surface structure methods are well documented and reproducible, transfer learning could be employed on a spectral library [28]. Ideally, researchers could contribute to this library in an open-access manner and use these spectra for model training. In this case, the quality of the attached metadata would be a crucial factor.

Clearly, machine learning has been used extensively in the context of SERS sensors. The most common pipeline is to perform unsupervised dimensionality reduction/feature extraction for which PCA is generally the preferred method. Less consistency is seen in the algorithms used for classification and regression. Alternatively, ANNs can be used directly on the data, and the advantage of one approach over the other is not clearly illustrated in the literature. We anticipate, however, that like in the case of electrochemical sensors, node-based models would allow for more efficient transfer learning to accommodate target task change.

5.4. Summary of Optical Bioreceptor-Free Biosensing

A variety of optical sensing methods have benefited from machine learning techniques, with the preferred method being dependent on the data type. For image type data, CNN is the most obvious choice for its ability to detect features as well as reconstruct images obtained by lensless systems. For spectral data, the approach is similar to spectral data obtained with electrochemical sensors. In those instances, dimensionality reduction coupled with a classification/regression algorithm may perform nearly as well as node-based methods. Indeed, they may be preferable in instances where the quantity of training data is small.

6. Considerations and Future Perspectives

Biosensor research has shown great success and promise. For both systems with and without bioreceptor, ML has demonstrated huge success in going from large, complex sensor datasets to getting meaningful measurements and classification of analytes. However, in many of these systems, a key challenge is consistency in device manufacturing. This manifests itself regarding sensor reproducibility for Enose and Etongue, or as substrate reproducibility for SERS. Since the models used to process these data often rely on subtle signals in the data, even small changes in sensor response characteristics can lead to poor performance. These issues have effectively limited widespread commercial adoption of these technologies. There has been some success in accommodating these inconsistencies through computational methods, notably with transfer learning for Enose. More work, both from a manufacturing and computational standpoint, needs to be done before many of these systems are robust enough for widespread adoption.

One area in which these systems have pushed to increase commercial potential is through miniaturization and modularity. There have been efforts with several of the methods presented here to develop compact standalone devices that rival their bulkier counterparts in terms of performance [16,47,162–166]. We believe that cloud computing may be a key element to the

success of these endeavors. Some of the models in use, especially for image-based sensors, are computationally expensive. By offloading the computational work to cloud computing, the device footprint imposed by processing and memory needs is greatly reduced.

A central question is what the relative advantages and disadvantages are between systems that utilize a bioreceptor and those that do not. A key advantage of those that eliminate the bioreceptor addresses one of the barriers to commercialization—manufacture variability. By eliminating the bioreceptor, device manufacture is simplified, and may decrease manufacture variability. Additionally, sensor longevity is generally improved because the long-term stability of the bioreceptor is often limited [6]. However, to match LOD and specificity of bioreceptors, improvements must be made. Nanomaterials show promise for improving device performance [167].

There have been studies that attempt to gain the advantages of both systems by creating artificial bioreceptors, notably nanomaterials with enzymatic properties referred to as nanozymes [168,169]. While exciting progress has been made in this field, current nanozyme-based biosensors have inferior catalytic activity and specificity to their biological alternatives [170,171]. Nanozyme catalytic activity is also currently limited to oxidase-like activity [171]. If researchers can broaden nanozyme activity and improve selectivity, these biosensors may become a competitive alternative for biological bioreceptors.

In addition to device considerations, there are computational challenges to consider. Although some ML algorithms have been in use for decades such as PCA and SVM, the field of ML is advancing rapidly with new algorithms being described frequently. While many areas are quick to adopt the new methods, improper usage is common and certainly not limited to biosensing. Some common mistakes are inappropriate data splitting, hidden variables serving as

bad predictors, and mistaking the objective of the model [172]. Great emphasis must be placed on the importance of reporting appropriate performance metrics. A great example of a misleading metric is reporting accuracy on highly imbalanced data such as in Durante et al. [41]. It can often be difficult to determine if the proper pre-processing and model assumption checks are being performed. This may be centering and re-scaling prior to PCA, or normality checks for LDA.

Some of these issues can be solved with better methods reporting, especially regarding computational methods. Certain key details are frequently left out, making critical evaluation difficult and reproducibility impossible [173]. One of the most striking examples from the literature described herein is reporting classification metrics, without reporting what classifier was used on PCA processed data [49]. Perhaps the best way to make methods clear and reproducible is to release all associated code, preferably publicly.

Increased availability in general can greatly improve this field. More open access repositories of training sets may allow researchers to improve model robustness by exposing them to more diverse datasets [16]. Some examples currently exist such as the gas sensor drift dataset [115] and the EIS breast tissue dataset [103], both available in the UCI Machine Learning repository [104]. One vision would be to have large repositories of gas sensor responses to many analytes under various experimental conditions. Models could be trained on such repositories to improve generalizability. Ideally, with such repositories and improved manufacturing consistency, trained models could be shared directly and need only minimal recalibration.

7. Conclusions

In this review we have explored the ways in which bioreceptor-free biosensors can benefit from ML methods. Robust ML models bring specificity and accuracy to array-based biosensors such as Enose and Etongue by learning the patterns in the sensor responses. Notably, PCA has

shown great performance as a feature extraction technique for these systems. Similar power of PCA has been demonstrated for optical biosensors that generate spectra such as Raman spectra or SERS. ANNs using deep learning generate impressive results for imaging-based sensors including lensless holography and digital staining. ML has also been used in creative ways such as for data fusion of multiple biosensors, and transfer learning for noise correction, sensor drift compensation, and domain adaptation.

However, many practical challenges still exist. Many of the methods presented here are not widely used in commercial settings. This is due to many reasons including variability in manufacturing and the ability to make compact versions of the biosensors while maintaining performance. ML models that can adapt to differences in sensor response are at an advantage, and transfer learning shows promise to be part of the solution.

In recent years, ML has garnered strong research interest in many fields including biosensing, as evidenced in this review. If this review has inspired interest to learn more about how machine learning is being used for one of the methods presented here, we encourage you to seek more specific reviews for the subject. There are great reviews in the literature, many of which were referenced, that take a closer look at the methods presented in this review.

8. Author Contributions

Conceptualization, K.E.S.III and J.-Y.Y.; Methodology, K.E.S.III; Formal analysis, K.E.S.III; Investigation, K.E.S.III; Data curation, K.E.S.III; Writing—original draft preparation, K.E.S.III; Writing—review and editing, J.-Y.Y.; Supervision, J.-Y.Y.; Project administration, J.-Y.Y.; Funding acquisition, K.E.S.III and J.-Y.Y. All authors have read and agreed to the published version of the manuscript.

9. Funding

This work was supported by the U.S. National Institutes of Health under the grant T32GM132008.

10. Acknowledgements

The authors would like to thank Lane E. Breshears for her contribution to the collection of papers and for discussions on how to organize this manuscript. The authors would also like to thank Kattika Kaarj (now at Mahidol University) for helpful discussions and proofreading this manuscript.

11. Conflicts of Interest

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

12. References

1. Metkar, S.K.; Girigoswami, K. Diagnostic biosensors in medicine—A review. Biocatal. Agric. Biotechnol. 2019, 17, 271–283, https://doi.org/1016/j.bcab.2018.11.029.
2. Justino, C.I.L.; Duarte, A.C.; Rocha-Santos, T.A.P. Recent progress in biosensors for environmental monitoring: A review. Sensors 2017, 17, 2918, https://doi.org/10.3390/s17122918.
3. Nguyen, H.H.; Lee, S.H.; Lee, U.J.; Fermin, C.D.; Kim, M. Immobilized enzymes in biosensor applications. Materials 2019, 12, 121, https://doi.org/10.3390/ma12010121.
4. Hock, B. Antibodies for immunosensors a review. Anal. Chim. Acta 1997, 347, 177–186, https://doi.org/10.1016/S0003-2670(97)00167-0.

5. Lim, Y.C.; Kouzani, A.Z.; Duan, W. Aptasensors: A review. J. Biomed. Nanotechnol. 2010, 6, 93–105, https://doi.org/10.1166/jbn.2010.1103.
6. Massah, J.; Asefpour Vakilian, K. An intelligent portable biosensor for fast and accurate nitrate determination using cyclic voltammetry. Biosyst. Eng. 2019, 177, 49–58, https://doi.org/10.1016/j.biosystemseng.2018.09.007.
7. Esfahani, S.; Shanta, M.; Specht, J.P.; Xing, Y.; Cole, M.; Gardner, J.W. Wearable IoT electronic nose for urinary incontinence detection. In Proceedings of the 2020 IEEE Sensors, Virtual Conference, 25–28 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4, https://doi.org/10.1109/SENSORS47125.2020.9278882
8. Pelosi, P.; Zhu, J.; Knoll, W. From gas sensors to biomimetic artificial noses. Chemosensors 2018, 6, 32, https://doi.org/10.3390/chemosensors6030032.
9. Wilson, A.D. Noninvasive early disease diagnosis by electronic-nose and related VOC-detection devices. Biosensors 2020, 10, 73, https://doi.org/10.3390/bios10070073.
10. Szulczyński, B.; Armiński, K.; Namieśnik, J.; Gębicki, J. Determination of odour interactions in gaseous mixtures using electronic nose methods with artificial neural networks. Sensors 2018, 18, 519, https://doi.org/10.3390/s18020519.
11. Zambotti, G.; Soprani, M.; Gobbi, E.; Capuano, R.; Pasqualetti, V.; Di Natale, C.; Ponzoni, A. Early detection of fish degradation by electronic nose. In Proceedings of the 2019 IEEE International Symposium on Olfaction and Electronic Nose (ISOEN), Fukuoka, Japan, 26–29 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–3, https://doi.org/10.1109/ISOEN.2019.8823461.
12. Podrażka, M.; Bączyńska, E.; Kundys, M.; Jeleń, P.S.; Witkowska Nery, E. Electronic tongue—A tool for all tastes? Biosensors 2018, 8, 3, https://doi.org/10.3390/bios8010003.

13. Chen, X.; Xu, Y.; Meng, L.; Chen, X.; Yuan, L.; Cai, Q.; Shi, W.; Huang, G. Non-parametric partial least squares—Discriminant analysis model based on sum of ranking difference algorithm for tea grade identification using electronic tongue data. Sens. Actuat. B Chem. 2020, 311, 127924, https://doi.org/10.1016/j.snb.2020.127924.
14. Kovacs, Z.; Szöllősi, D.; Zaukuu, J.-L.Z.; Bodor, Z.; Vitális, F.; Aouadi, B.; Zsom-Muha, V.; Gillay, Z. Factors influencing the long-term stability of electronic tongue and application of improved drift correction methods. Biosensors 2020, 10, 74, https://doi.org/10.3390/bios10070074.
15. Guedes, M.D.V.; Marques, M.S.; Guedes, P.C.; Contri, R.V.; Kulkamp Guerreiro, I.C. The use of electronic tongue and sensory panel on taste evaluation of pediatric medicines: A systematic review. Pharm. Dev. Technol. 2020, 26, 119–137, https://doi.org/10.1080/10837450.2020.1860088.
16. Ross, C.F. Considerations of the use of the electronic tongue in sensory science. Curr. Opin. Food Sci. 2021, 40, 87–93, https://doi.org/10.1016/j.cofs.2021.01.011.
17. Guerrini, L.; Alvarez-Puebla, R.A. Chapter 19—Surface-enhanced Raman scattering chemosensing of proteins. In Vibrational Spectroscopy in Protein Research; Ozaki, Y., Baranska, M., Lednev, I.K., Wood, B.R., Eds.; Academic Press: London, UK, 2020; pp. 553–567, https://doi.org/10.1016/B978-0-12-818610-7.00019-0.
18. Feng, J.; Hu, Y.; Grant, E.; Lu, X. Determination of thiabendazole in orange juice using an MISPE-SERS chemosensor. Food Chem. 2018, 239, 816–822, https://doi.org/10.1016/j.foodchem.2017.07.014.
19. Langer, J.; Jimenez de Aberasturi, D.; Aizpurua, J.; Alvarez-Puebla, R.A.; Auguié, B.; Baumberg, J.J.; Bazan, G.C.; Bell, S.E.J.; Boisen, A.; Brolo, A.G.; et al. Present and future of

surface-enhanced Raman scattering. ACS Nano 2020, 14, 28–117, https://doi.org/10.1021/acsnano.9b04224.

20. Zheng, X.S.; Jahn, I.J.; Weber, K.; Cialla-May, D.; Popp, J. Label-free SERS in biological and biomedical applications: Recent progress, current challenges and opportunities. Spectrochim. Acta A Mol. Biomol. Spectrosc. 2018, 197, 56–77, https://doi.org/10.1016/j.saa.2018.01.063.
21. Krafft, C.; Osei, E.B.; Popp, J.; Nazarenko, I. Raman and SERS spectroscopy for characterization of extracellular vesicles from control and prostate carcinoma patients. Proc. SPIE 2020, 11236, 1123602, https://doi.org/10.1117/12.2549209.
22. Sang, S.; Wang, Y.; Feng, Q.; Wei, Y.; Ji, J.; Zhang, W. Progress of new label-free techniques for biosensors: A review. Crit. Rev. Biotechnol. 2016, 36, 465–481, https://doi.org/10.3109/07388551.2014.991270.
23. Scott, S.M.; James, D.; Ali, Z. Data analysis for electronic nose systems. Microchim. Acta 2006, 156, 183–207, https://doi.org/10.1007/s00604-006-0623-9.
24. Yan, J.; Guo, X.; Duan, S.; Jia, P.; Wang, L.; Peng, C.; Zhang, S. Electronic nose feature extraction methods: A review. Sensors 2015, 15, 27804–27831, https://doi.org/10.3390/s151127804.
25. Hotel, O.; Poli, J.-P.; Mer-Calfati, C.; Scorsone, E.; Saada, S. A review of algorithms for SAW sensors Enose based volatile compound identification. Sens. Actuat. B Chem. 2018, 255, 2472–2482, https://doi.org/10.1016/j.snb.2017.09.040.
26. da Costa, N.L.; da Costa, M.S.; Barbosa, R. A review on the application of chemometrics and machine learning algorithms to evaluate beer authentication. Food Anal. Meth. 2021, 14, 136–155, https://doi.org/10.1007/s12161-020-01864-7.

27. Wasilewski, T.; Migoń, D.; Gębicki, J.; Kamysz, W. Critical review of electronic nose and tongue instruments prospects in pharmaceutical analysis. Anal. Chim. Acta 2019, 1077, 14-29, https://doi.org/10.1016/j.aca.2019.05.024.
28. Lussier, F.; Thibault, V.; Charron, B.; Wallace, G.Q.; Masson, J.-F. Deep learning and artificial intelligence methods for Raman and surface-enhanced Raman scattering. TrAC-Trends Anal. Chem. 2020, 124, 115796, https://doi.org/10.1016/j.trac.2019.115796.
29. Cui, F.; Yue, Y.; Zhang, Y.; Zhang, Z.; Zhou, H.S. Advancing biosensors with machine learning. ACS Sens. 2020, 5, 3346–3364, https://doi.org/10.1021/acssensors.0c01424.
30. Sheng, Y.; Qian, W.; Huang, J.; Wu, B.; Yang, J.; Xue, T.; Ge, Y.; Wen, Y. Electrochemical detection combined with machine learning for intelligent sensing of maleic hydrazide by using carboxylated PEDOT modified with copper nanoparticles. Microchim. Acta 2019, 186, 543, https://doi.org/10.1007/s00604-019-3652-x.
31. Dean, S.N.; Shriver-Lake, L.C.; Stenger, D.A.; Erickson, J.S.; Golden, J.P.; Trammell, S.A. Machine learning techniques for chemical identification using cyclic square wave voltammetry. Sensors 2019, 19, 2392, https://doi.org/10.3390/s19102392.
32. Tonezzer, M.; Le, D.T.T.; Iannotta, S.; Van Hieu, N. Selective discrimination of hazardous gases using one single metal oxide resistive sensor. Sens. Actuat. B Chem. 2018, 277, 121–128, https://doi.org/10.1016/j.snb.2018.08.103.
33. Xu, L.; He, J.; Duan, S.; Wu, X.; Wang, Q. Comparison of machine learning algorithms for concentration detection and prediction of formaldehyde based on electronic nose. Sens. Rev. 2016, 36, 207–216, https://doi.org/10.1108/SR-07-2015-0104.

34. Yang, Y.; Liu, H.; Gu, Y. A model transfer learning framework with back-propagation neural network for wine and Chinese liquor detection by electronic nose. IEEE Access 2020, 8, 105278–105285, https://doi.org/10.1109/ACCESS.2020.2999591.
35. Leon-Medina, J.X.; Pineda-Muñoz, W.A.; Burgos, D.A.T. Joint distribution adaptation for drift correction in electronic nose type sensor arrays. IEEE Access 2020, 8, 134413–134421, https://doi.org/10.1109/ACCESS.2020.3010711.
36. Liu, B.; Zeng, X.; Tian, F.; Zhang, S.; Zhao, L. Domain transfer broad learning system for long-term drift compensation in electronic nose systems. IEEE Access 2019, 7, 143947–143959, https://doi.org/10.1109/ACCESS.2019.2943188.
37. Wang, X.; Gu, Y.; Liu, H. A transfer learning method for the protection of geographical indication in China using an electronic nose for the identification of Xihu Longjing tea. IEEE Sens. J. 2021, 21, 8065–8077, https://doi.org/10.1109/JSEN.2020.3048534.
38. Yi, R.; Yan, J.; Shi, D.; Tian, Y.; Chen, F.; Wang, Z.; Duan, S. Improving the performance of drifted/shifted electronic nose systems by cross-domain transfer using common transfer samples. Sens. Actuat. B Chem. 2021, 329, 129162, https://doi.org/10.1016/j.snb.2020.129162.
39. Liang, Z.; Tian, F.; Zhang, C.; Sun, H.; Song, A.; Liu, T. Improving the robustness of prediction model by transfer learning for interference suppression of electronic nose. IEEE Sens. J. 2018, 18, 1111–1121, https://doi.org/10.1109/JSEN.2017.2778012.
40. Daliri, M.R. Combining extreme learning machines using support vector machines for breast tissue classification. Comput. Meth. Biomech. Biomed. Eng. 2015, 18, 185–191, https://doi.org/10.1080/10255842.2013.789100.

41. Durante, G.; Becari, W.; Lima, F.A.S.; Peres, H.E.M. Electrical impedance sensor for real-time detection of bovine milk adulteration. IEEE Sens. J. 2016, 16, 861–865, https://doi.org/10.1109/JSEN.2015.2494624.
42. Helwan, A.; Idoko, J.B.; Abiyev, R.H. Machine learning techniques for classification of breast tissue. Proc. Comput. Sci. 2017, 120, 402–410, https://doi.org/10.1016/j.procs.2017.11.256.
43. Islam, M.; Wahid, K.; Dinh, A. Assessment of ripening degree of avocado by electrical impedance spectroscopy and support vector machine. J. Food Qual. 2018, 2018, 4706147, https://doi.org/10.1155/2018/4706147.
44. Murphy, E.K.; Mahara, A.; Khan, S.; Hyams, E.S.; Schned, A.R.; Pettus, J.; Halter, R.J. Comparative study of separation between ex vivo prostatic malignant and benign tissue using electrical impedance spectroscopy and electrical impedance tomography. Physiol. Meas. 2017, 38, 1242–1261, https://doi.org/10.1088/1361-6579/aa660e.
45. Yang, Z.; Miao, N.; Zhang, X.; Li, Q.; Wang, Z.; Li, C.; Sun, X.; Lan, Y. Employment of an electronic tongue combined with deep learning and transfer learning for discriminating the storage time of Pu-erh tea. Food Control. 2021, 121, 107608, https://doi.org/10.1016/j.foodcont.2020.107608.
46. Leon-Medina, J.X.; Anaya, M.; Pozo, F.; Tibaduiza, D. Nonlinear feature extraction through manifold learning in an electronic tongue classification task. Sensors 2020, 20, 4834, https://doi.org/10.3390/s20174834.
47. Giménez-Gómez, P.; Escudé-Pujol, R.; Capdevila, F.; Puig-Pujol, A.; Jiménez-Jorquera, C.; Gutiérrez-Capitán, M. Portable electronic tongue based on microsensors for the analysis of Cava wines. Sensors 2016, 16, 1796, https://doi.org/10.3390/s16111796.

48. Ouyang, Q.; Yang, Y.; Wu, J.; Liu, Z.; Chen, X.; Dong, C.; Chen, Q.; Zhang, Z.; Guo, Z. Rapid sensing of total theaflavins content in black tea using a portable electronic tongue system coupled to efficient variables selection algorithms. J. Food Compos. Anal. 2019, 75, 43–48, https://doi.org/10.1016/j.jfca.2018.09.014.
49. Li, Z.; Paul, R.; Ba Tis, T.; Saville, A.C.; Hansel, J.C.; Yu, T.; Ristaino, J.B.; Wei, Q. Non-invasive plant disease diagnostics enabled by smartphone-based fingerprinting of leaf volatiles. Nat. Plants 2019, 5, 856–866, https://doi.org/10.1038/s41477-019-0476-y.
50. Tomczak, A.; Ilic, S.; Marquardt, G.; Engel, T.; Forster, F.; Navab, N.; Albarqouni, S. Multi-task multi-domain learning for digital staining and classification of leukocytes. IEEE Trans. Med. Imaging 2020, https://doi.org/10.1109/TMI.2020.3046334, https://doi.org/10.1109/TMI.2020.3046334.
51. Sagar, M.A.K.; Cheng, K.P.; Ouellette, J.N.; Williams, J.C.; Watters, J.J.; Eliceiri, K.W. Machine learning methods for fluorescence lifetime imaging (FLIM) based label-free detection of microglia. Front. Neurosci. 2020, 14, 931, https://doi.org/10.3389/fnins.2020.00931.
52. Lotfollahi, M.; Berisha, S.; Daeinejad, D.; Mayerich, D. Digital staining of high-definition Fourier transform infrared (FT-IR) images using deep learning. Appl. Spectrosc. 2019, 73, 556–564, https://doi.org/10.1177/0003702818819857.
53. Rivenson, Y.; Zhang, Y.; Günaydın, H.; Teng, D.; Ozcan, A. Phase recovery and holographic image reconstruction using deep learning in neural networks. Light Sci. Appl. 2018, 7, 17141, https://doi.org/10.1038/lsa.2017.141.
54. Rivenson, Y.; Wu, Y.; Ozcan, A. Deep learning in holography and coherent imaging. Light Sci. Appl. 2019, 8, 85, https://doi.org/10.1038/s41377-019-0196-0.

55. Wu, Y.; Ray, A.; Wei, Q.; Feizi, A.; Tong, X.; Chen, E.; Luo, Y.; Ozcan, A. Deep learning enables high-throughput analysis of particle-aggregation-based biosensors imaged using holography. ACS Photon. 2019, 6, 294–301, https://doi.org/10.1021/acsphotonics.8b01479.
56. Wu, Y.; Calis, A.; Luo, Y.; Chen, C.; Lutton, M.; Rivenson, Y.; Lin, X.; Koydemir, H.C.; Zhang, Y.; Wang, H.; Göröcs, Z.; Ozcan, A. Label-free bioaerosol sensing using mobile microscopy and deep learning. ACS Photon. 2018, 5, 4617–4627, https://doi.org/10.1021/acsphotonics.8b01109.
57. Borhani, N.; Bower, A.J.; Boppart, S.A.; Psaltis, D. Digital staining through the application of deep neural networks to multi-modal multi-photon microscopy. Biomed. Opt. Express 2019, 10, 1339–1350, https://doi.org/10.1364/BOE.10.001339.
58. Dunker, S.; Motivans, E.; Rakosy, D.; Boho, D.; Mäder, P.; Hornick, T.; Knight, T.M. Pollen analysis using multispectral imaging flow cytometry and deep learning. New Phytol. 2021, 229, 593–606, https://doi.org/10.1111/nph.16882.
59. Rivenson, Y.; Liu, T.; Wei, Z.; Zhang, Y.; de Haan, K.; Ozcan, A. PhaseStain: The digital staining of label-free quantitative phase microscopy images using deep learning. Light Sci. Appl. 2019, 8, 23, https://doi.org/10.1038/s41377-019-0129-y.
60. Zheng, X.; Lv, G.; Du, G.; Zhai, Z.; Mo, J.; Lv, X. Rapid and low-cost detection of thyroid dysfunction using Raman spectroscopy and an improved support vector machine. IEEE Photon. J. 2018, 10, 1–12, https://doi.org/10.1109/JPHOT.2018.2876686.
61. Tan, A.; Zhao, Y.; Sivashanmugan, K.; Squire, K.; Wang, A.X. Quantitative TLC-SERS detection of histamine in seafood with support vector machine analysis. Food Control. 2019, 103, 111–118, https://doi.org/10.1016/j.foodcont.2019.03.032.

62. Chen, X.; Nguyen, T.H.D.; Gu, L.; Lin, M. Use of standing gold nanorods for detection of malachite green and crystal violet in fish by SERS. J. Food Sci. 2017, 82, 1640–1646, https://doi.org/10.1111/1750-3841.13766.

63. Fornasaro, S.; Marta, S.D.; Rabusin, M.; Bonifacio, A.; Sergo, V., Toward SERS-based point-of-care approaches for therapeutic drug monitoring: The case of methotrexate. Faraday Discuss. 2016, 187, 485–499, https://doi.org/10.1039/C5FD00173K.

64. Hassoun, M.; Rüger, J.; Kirchberger-Tolstik, T.; Schie, I.W.; Henkel, T.; Weber, K.; Cialla-May, D.; Krafft, C.; Popp, J. A droplet-based microfluidic chip as a platform for leukemia cell lysate identification using surface-enhanced Raman scattering. Anal. Bioanal. Chem. 2018, 410, 999–1006, https://doi.org/10.1007/s00216-017-0609-y.

65. Hidi, I.J.; Jahn, M.; Pletz, M.W.; Weber, K.; Cialla-May, D.; Popp, J. Toward levofloxacin monitoring in human urine samples by employing the LoC-SERS technique. J. Phys. Chem. C 2016, 120, 20613–20623, https://doi.org/10.1021/acs.jpcc.6b01005.

66. Hou, M.; Huang, Y.; Ma, L.; Zhang, Z. Quantitative analysis of single and mix food antiseptics basing on SERS spectra with PLSR method. Nanoscale Res. Lett. 2016, 11, 296, https://doi.org/10.1186/s11671-016-1507-5.

67. Kämmer, E.; Olschewski, K.; Stöckel, S.; Rösch, P.; Weber, K.; Cialla-May, D.; Bocklitz, T.; Popp, J. Quantitative SERS studies by combining LOC-SERS with the standard addition method. Anal. Bioanal. Chem. 2015, 407, 8925–8929, https://doi.org/10.1007/s00216-015-9045-z.

68. Mühlig, A.; Bocklitz, T.; Labugger, I.; Dees, S.; Henk, S.; Richter, E.; Andres, S.; Merker, M.; Stöckel, S.; Weber, K.; Cialla-May, D.; Popp, J. LOC-SERS: A promising closed system

for the identification of mycobacteria. Anal. Chem. 2016, 88, 7998–8004, https://doi.org/10.1021/acs.analchem.6b01152.

69. Nguyen, C.Q.; Thrift, W.J.; Bhattacharjee, A.; Ranjbar, S.; Gallagher, T.; Darvishzadeh-Varcheie, M.; Sanderson, R.N.; Capolino, F.; Whiteson, K.; Baldi, P.; Hochbaum, A.I.; Ragan, R. Longitudinal monitoring of biofilm formation via robust surface-enhanced Raman scattering quantification of Pseudomonas aeruginosa-produced metabolites. ACS Appl. Mater. Interfaces 2018, 10, 12364–12373, https://doi.org/10.1021/acsami.7b18592.
70. Othman, N.H.; Lee, K.Y.; Radzol, A.R.M.; Mansor, W. PCA-SCG-ANN for detection of non-structural protein 1 from SERS salivary spectra. In: Intelligent Information and Database Systems; Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B., Eds.; Springer: Cham, Germany, 2017; pp. 424–433, https://doi.org/10.1007/978-3-319-54430-4_41.
71. Othman, N.H.; Yoot Lee, K.; Mohd Radzol, A.R.; Mansor, W.; Amanina Yusoff, N. PCA-polynomial-ELM model optimal for detection of NS1 adulterated salivary SERS spectra. J. Phys. Conf. Ser. 2019, 1372, 012064, https://doi.org/10.1088/1742-6596/1372/1/012064.
72. Seifert, S.; Merk, V.; Kneipp, J. Identification of aqueous pollen extracts using surface enhanced Raman scattering (SERS) and pattern recognition methods. J. Biophoton. 2016, 9, 181–189, https://doi.org/10.1002/jbio.201500176.
73. Sun, H.; Lv, G.; Mo, J.; Lv, X.; Du, G.; Liu, Y. Application of KPCA combined with SVM in Raman spectral discrimination. Optik 2019, 184, 214–219, https://doi.org/10.1016/j.ijleo.2019.02.126.
74. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. Chemom. Intell. Lab. Syst. 1987, 2, 37–52, https://doi.org/10.1016/0169-7439(87)80084-9.

75. Cunningham, P. Dimension reduction. In Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval, Cognitive Technologies; Cord, M., Cunningham, P., Eds.; Springer: Berlin-Heidelberg, Germany, 2008; pp. 91–112, https://doi.org/10.1007/978-3-540-75171-7_4.
76. Alpaydin, E. Introduction to Machine Learning; MIT Press: Cambridge, MA, USA, 2020.
77. Peterson, L.E. K-nearest neighbor. Scholarpedia 2009, 4, 1883, https://doi.org/10.4249/scholarpedia.1883.
78. Balaprakash, P.; Salim, M.; Uram, T.D.; Vishwanath, V.; Wild, S.M. DeepHyper: Asynchronous hyperparameter search for deep neural networks. In Proceedings of the 2018 IEEE 25th International Conference on High Performance Computing (HiPC), Bengaluru, India, 17–20 December 2018; IEEE: Piscataway, NJ, USA, 2019; pp. 42–51, https://doi.org/10.1109/HiPC.2018.00014.
79. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT'92, Pittsburgh, PA, USA, 27–29 July 1992; Association for Computing Machinery: New York, NY, USA, 1992; pp. 144–152, https://doi.org/10.1145/130385.130401.
80. Zhang, J. A complete list of kernels used in support vector machines. Biochem. Pharmacol. 2015, 4, 2, https://doi.org/10.4172/2167-0501.1000195.
81. Mathur, A.; Foody, G.M. Multiclass and binary SVM classification: Implications for training and classification users. IEEE Geosci. Remote Sens. Lett. 2008, 5, 241–245, https://doi.org/10.1109/LGRS.2008.915597.
82. Shawe-Taylor, J.; Cristianini, N. On the generalization of soft margin algorithms. IEEE Trans. Inf. Theory 2002, 48, 2721–2735, https://doi.org/10.1109/TIT.2002.802647.

83. Han, H.; Jiang, X. Overcome support vector machine diagnosis overfitting. Cancer Inform. 2014, 13, 145-158, https://doi.org/10.4137/CIN.S13875.
84. Ghojogh, B.; Crowley, M. Linear and quadratic discriminant analysis: Tutorial. arXiv 2019, arXiv:1906.02590. Available online: https://arxiv.org/abs/1906.02590v1. Accessed on 2 August, 2021.
85. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An introduction to decision tree modeling. J. Chemom. 2004, 18, 275–285, https://doi.org/10.1002/cem.873.
86. Lewis, R.J. An introduction to classification and regression tree (CART) analysis. In The 2000 Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California; 2000. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.4103&rep=rep1&type=pdf. Accessed 4 Aug, 2021.
87. Criminisi, A.; Shotton, J.; Konukoglu, E. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found. Trends. Comput. Graph. Vis. 2012, 7, 81–227, https://doi.org/10.1561/0600000035.
88. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys. D Nonlinear Phenom. 2020, 404, 132306, https://doi.org/10.1016/j.physd.2019.132306.
89. Huang, G.B.; Zhu, Q.Y.; Siew, C.-K. Extreme learning machine: Theory and applications. Neurocomput. Neural Netw. 2006, 70, 489–501, https://doi.org/10.1016/j.neucom.2005.12.126.
90. Fukushima, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural Netw. 1998, 1, 119–130, https://doi.org/10.1016/0893-6080(88)90014-7.

91. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. Science 2006, 313, 504–507, https://doi.org/10.1126/science.1127647.
92. Hecht-Nielsen, R. Theory of the backpropagation neural network. In Neural Networks for Perception, Volume 2: Computation, Learning, and Architectures; Academic Press: Boston, MA, USA, 1992; pp. 65–93, https://doi.org/10.1016/B978-0-12-741252-8.50010-8.
93. Alto, V. Neural networks: Parameters, hyperparameters and optimization strategies. Available online: https://towardsdatascience.com/neural-networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5 (accessed on 2 August 2021).
94. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. arXiv 2014, arXiv:1406.2661. Available online: https://arxiv.org/abs/1406.2661. Accessed 13 April, 2021.
95. Zhang, F.; O'Donnell, L.J. Chapter 7—Support vector regression. In Machine Learning; Mechelli, A., Vieira, S., Eds.; Academic Press: London, UK, 2020; pp. 123–140, https://doi.org/10.1016/B978-0-12-815739-8.00007-9.
96. Hoffmann, F.; Bertram, T.; Mikut, R.; Reischl, M.; Nelles, O. Benchmarking in classification and regression. WIREs 2019, 9, e1318, https://doi.org/10.1002/widm.1318.
97. Shao, J. Linear model selection by cross-validation. J. Am. Stat. Assoc. 1993, 88, 486–494, https://doi.org/10.1080/01621459.1993.10476299.
98. Yoon, J.-Y. Introduction to Biosensors: From Electric Circuits to Immunosensors, 2nd ed.; Springer: New York: NY, USA, 2016, https://doi.org/10.1007/978-3-319-27413-3.
99. Grieshaber, D.; MacKenzie, R.; Vörös, J.; Reimhult, E. Electrochemical biosensors—Sensor principles and architectures. Sensors 2008, 8, 1400–1458, https://doi.org/10.3390/s80314000.

100. Dhanjai; Sinha, A.; Lu, X.; Wu, L.; Tan, D.; Li, Y.; Chen, J.; Jain, R. Voltammetric sensing of biomolecules at carbon based electrode interfaces: A review. TrAC—Trends Anal. Chem. 2018, 98, 174–189, https://doi.org/10.1016/j.trac.2017.11.010.

101. Grossi, M.; Riccò, B. Electrical impedance spectroscopy (EIS) for biological analysis and food characterization: A review. J. Sens. Sens. Syst. 2017, 6, 303–325, https://doi.org/10.5194/jsss-6-303-2017.

102. Yao, J.; Wang, L.; Liu, K.; Wu, H.; Wang, H.; Huang, J.; Li, J. Evaluation of electrical characteristics of biological tissue with electrical impedance spectroscopy. Electrophoresis 2020, 41, 1425–1432, https://doi.org/10.1002/elps.201900420.

103. Jossinet, J. Variability of impedivity in normal and pathological breast tissue. Med. Biol. Eng. Comput. 1996, 34, 346–350, https://doi.org/10.1007/BF02520002.

104. Dua, D.; Graff, C. UCI Machine Learning Repository (https://archive.ics.uci.edu/); University of California, Irvine, School of Information and Computer Sciences: Irvine, CA, USA, 2017.

105. Estrela da Silva, J.; Marques de Sá, J.P.; Jossinet, J. Classification of breast tissue by electrical impedance spectroscopy. Med. Biol. Eng. Comput. 2000, 38, 26–30, https://doi.org/10.1007/BF02344684.

106. Wasilewski, T.; Kamysz, W.; Gębicki, J. Bioelectronic tongue: Current status and perspectives. Biosens. Bioelectron. 2020, 150, 111923, https://doi.org/10.1016/j.bios.2019.111923.

107. Röck, F.; Barsan, N.; Weimar, U. Electronic nose: Current status and future trends. Chem. Rev. 2008, 108, 705–725, https://doi.org/10.1021/cr068121q.

108. Ha, D.; Sun, Q.; Su, K.; Wan, H.; Li, H.; Xu, N.; Sun, F.; Zhuang, L.; Hu, N.; Wang, P. Recent achievements in electronic tongue and bioelectronic tongue as taste sensors. Sens. Actuat. B: Chem. 2015, 207, 1136–1146, https://doi.org/10.1016/j.snb.2014.09.077.

109. Chen, C.Y.; Lin, W.C.; Yang, H.Y. Diagnosis of ventilator-associated pneumonia using electronic nose sensor array signals: Solutions to improve the application of machine learning in respiratory research. Respir. Res. 2020, 21, 45, https://doi.org/10.1186/s12931-020-1285-6.

110. Tan, J.; Xu, J. Applications of electronic nose (Enose) and electronic tongue (Etongue) in food quality-related properties determination: A review. Artif. Intell. Agric. 2020, 4, 104–115, https://doi.org/10.1016/j.aiia.2020.06.003.

111. Sanaeifar, A.; ZakiDizaji, H.; Jafari, A.; de la Guardia, M. Early detection of contamination and defect in foodstuffs by electronic nose: A review. TrAC—Trends Anal. Chem. 2017, 97, 257–271, https://doi.org/10.1016/j.trac.2017.09.014.

112. Zhang, L.; Tian, F.; Kadri, C.; Xiao, B.; Li, H.; Pan, L.; Zhou, H. On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality. Sens. Actuat. B Chem. 2011, 160, 899–909, https://doi.org/10.1016/j.snb.2011.08.079.

113. Ciosek, P.; Wróblewski, W. Sensor arrays for liquid sensing—Electronic tongue systems. Analyst 2007, 132, 963–978, https://doi.org/10.1039/B705107G.

114. Liu, H.; Li, Q.; Li, Z.; Gu, Y. A suppression method of concentration background noise by transductive transfer learning for a metal oxide semiconductor-based electronic nose. Sensors 2020, 20, 1913, https://doi.org/10.3390/s20071913.

115. Vergara, A.; Vembu, S.; Ayhan, T.; Ryan, M.A.; Homer, M.L.; Huerta, R. Chemical gas sensor drift compensation using classifier ensembles. Sens. Actuat. B Chem. 2012, 166–167, 320–329, https://doi.org/10.1016/j.snb.2012.01.074.

116. Zhang, L.; Tian, F.; Peng, X.; Dang, L.; Li, G.; Liu, S.; Kadri, C. Standardization of metal oxide sensor array using artificial neural networks through experimental design. Sens. Actuat. B Chem. 2013, 177, 947–955, https://doi.org/10.1016/j.snb.2012.11.113.

117. DeVere, R. Disorders of taste and smell. Continuum 2017, 23, 421, https://doi.org/10.1212/CON.0000000000000463.

118. Biancolillo, A.; Boqué, R.; Cocchi, M.; Marini, F. Chapter 10—Data fusion strategies in food analysis. In Data Handling in Science and Technology; Cocchi, M., Ed.; Elsevier: Amsterdam, Netherlands, 2019; Volume 31, pp. 271–310, https://doi.org/10.1016/B978-0-444-63984-4.00010-7.

119. Banerjee, M.B.; Roy, R.B.; Tudu, B.; Bandyopadhyay, R.; Bhattacharyya, N. Black tea classification employing feature fusion of E-nose and E-tongue responses. J. Food Eng. 2019, 244, 55–63, https://doi.org/10.1016/j.jfoodeng.2018.09.022.

120. Men, H.; Shi, Y.; Fu, S.; Jiao, Y.; Qiao, Y.; Liu, J. Mining feature of data fusion in the classification of beer flavor information using E-tongue and E-nose. Sensors 2017, 17, 1656, https://doi.org/10.3390/s17071656.

121. Buratti, S.; Malegori, C.; Benedetti, S.; Oliveri, P.; Giovanelli, G. E-nose, E-tongue and E-eye for edible olive oil characterization and shelf life assessment: A powerful data fusion approach. Talanta 2018, 182, 131–141, https://doi.org/10.1016/j.talanta.2018.01.096.

122. Dai, C.; Huang, X.; Huang, D.; Lv, R.; Sun, J.; Zhang, Z.; Ma, M.; Aheto, J.H. Detection of submerged fermentation of Tremella aurantialba using data fusion of electronic nose and tongue. J. Food Proc. Eng. 2019, 42, e13002, https://doi.org/10.1111/jfpe.13002.

123. Tian, X.; Wang, J.; Ma, Z.; Li, M.; Wei, Z. Combination of an E-nose and an E-tongue for adulteration detection of minced mutton mixed with pork. J. Food Qual. 2019, 2019, e4342509, https://doi.org/10.1155/2019/4342509.

124. Di Rosa, A.R.; Leone, F.; Cheli, F.; Chiofalo, V. Fusion of electronic nose, electronic tongue and computer vision for animal source food authentication and quality assessment—A review. J. Food Eng. 2017, 210, 62–75, https://doi.org/10.1016/j.jfoodeng.2017.04.024.

125. Cave, J.W.; Wickiser, J.K.; Mitropoulos, A.N. Progress in the development of olfactory-based bioelectronic chemosensors. Biosens. Bioelectron. 2019, 123, 211–222, https://doi.org/10.1016/j.bios.2018.08.063.

126. Ahn, S.R.; An, J.H.; Song, H.S.; Park, J.W.; Lee, S.H.; Kim, J.H.; Jang, J.; Park, T.H. Duplex bioelectronic tongue for sensing umami and sweet tastes based on human taste receptor nanovesicles. ACS Nano 2016, 10, 7287–7296, https://doi.org/10.1021/acsnano.6b02547.

127. Edachana, R.P.; Kumaresan, A.; Balasubramanian, V.; Thiagarajan, R.; Nair, B.G.; Thekkedath Gopalakrishnan, S.B. Paper-based device for the colorimetric assay of bilirubin based on in-vivo formation of gold nanoparticles. Microchim. Acta 2019, 187, 60, https://doi.org/10.1007/s00604-019-4051-z.

128. Mutlu, A.Y.; Kılıç, V.; Kocakuşak Özdemir, G.; Bayram, A.; Horzum, N.; Solmaz, M.E., Smartphone-based colorimetric detection via machine learning. Analyst 2017, 142, 2434–2441, https://doi.org/10.1039/C7AN00741H.

129. Solmaz, M.E.; Mutlu, A.Y.; Alankus, G.; Kılıç, V.; Bayram, A.; Horzum, N. Quantifying colorimetric tests using a smartphone app based on machine learning classifiers. Sens. Actuat. B Chem. 2018, 255, 1967–1973, https://doi.org/10.1016/j.snb.2017.08.220.

130. Lin, B.; Yu, Y.; Cao, Y.; Guo, M.; Zhu, D.; Dai, J.; Zheng, M. Point-of-care testing for streptomycin based on aptamer recognizing and digital image colorimetry by smartphone. Biosens. Bioelectron. 2018, 100, 482–489, https://doi.org/10.1016/j.bios.2017.09.028.

131. Song, E.; Yu, M.; Wang, Y.; Hu, W.; Cheng, D.; Swihart, M.T.; Song, Y. Multi-color quantum dot-based fluorescence immunoassay array for simultaneous visual detection of multiple antibiotic residues in milk. Biosens. Bioelectron. 2015, 72, 320–325, https://doi.org/10.1016/j.bios.2015.05.018.

132. Wang, L.; Chen, W.; Ma, W.; Liu, L.; Ma, W.; Zhao, Y.; Zhu, Y.; Xu, L.; Kuang, H.; Xu, C. Fluorescent strip sensor for rapid determination of toxins. Chem. Commun. 2011, 47, 1574–1576, https://doi.org/10.1039/C0CC04032K.

133. Chen, X.; Lan, J.; Liu, Y.; Li, L.; Yan, L.; Xia, Y.; Wu, F.; Li, C.; Li, S.; Chen, J. A paper-supported aptasensor based on upconversion luminescence resonance energy transfer for the accessible determination of exosomes. Biosens. Bioelectron. 2018, 102, 582–588, https://doi.org/10.1016/j.bios.2017.12.012.

134. Guo, X. Surface plasmon resonance based biosensor technique: A review. J. Biophoton. 2012, 5, 483–501, https://doi.org/10.1002/jbio.201200015.

135. Heinze, B.C.; Yoon, J.-Y. Nanoparticle immunoagglutination Rayleigh scatter assay to complement microparticle immunoagglutination Mie scatter assay in a microfluidic device. Colloids Surf. B Biointerfaces 2011, 85, 168–173, https://doi.org/10.1016/j.colsurfb.2011.02.024.

136. Park, T.S.; Li, W.; McCracken, K.E.; Yoon, J.-Y. Smartphone quantifies Salmonella from paper microfluidics. Lab Chip 2013, 13, 4832–4840, https://doi.org/10.1039/C3LC50976A.

137. Elad, T.; Benovich, E.; Magrisso, S.; Belkin, S. Toxicant identification by a luminescent bacterial bioreporter panel: Application of pattern classification algorithms. Environ. Sci. Technol. 2008, 42, 8486–8491, https://doi.org/10.1021/es801489a.

138. Jouanneau, S.; Durand, M.-J.; Courcoux, P.; Blusseau, T.; Thouand, G. Improvement of the identification of four heavy metals in environmental samples by using predictive decision tree models coupled with a set of five bioluminescent bacteria. Environ. Sci. Technol. 2011, 45, 2925–2931, https://doi.org/10.1021/es1031757.

139. Gou, T.; Hu, J.; Zhou, S.; Wu, W.; Fang, W.; Sun, J.; Hu, Z.; Shen, H.; Mu, Y. A new method using machine learning for automated image analysis applied to chip-based digital assays. Analyst 2019, 144, 3274–3281, https://doi.org/10.1039/C9AN00149B.

140. Ker, J.; Wang, L.; Rao, J.; Lim, T. Deep learning applications in medical image analysis. IEEE Access 2018, 6, 9375–9389, https://doi.org/10.1109/ACCESS.2017.2788044.

141. Uslu, F.; Icoz, K.; Tasdemir, K.; Yilmaz, B. Automated quantification of immunomagnetic beads and leukemia cells from optical microscope images. Biomed. Signal Proc. Control. 2019, 49, 473–482, https://doi.org/10.1016/j.bspc.2019.01.002.

142. Zeng, N.; Wang, Z.; Zhang, H.; Liu, W.; Alsaadi, F.E. Deep belief networks for quantitative analysis of a gold immunochromatographic strip. Cogn. Comput. 2016, 8, 684–692, https://doi.org/10.1007/s12559-016-9404-x.

143. Roy, M.; Seo, D.; Oh, S.; Yang, J.-W.; Seo, S. A review of recent progress in lens-free imaging and sensing. Biosens. Bioelectron. 2017, 88, 130–143, https://doi.org/10.1016/j.bios.2016.07.115.

144. Wu, Y.; Ozcan, A. Lensless digital holographic microscopy and its applications in biomedicine and environmental monitoring. Methods 2018, 136, 4–16, https://doi.org/10.1016/j.ymeth.2017.08.013.

145. Greenbaum, A.; Luo, W.; Su, T.W.; Göröcs, Z.; Xue, L.; Isikman, S.O.; Coskun, A.F.; Mudanyali, O.; Ozcan, A. Imaging without lenses: Achievements and remaining challenges of wide-field on-chip microscopy. Nat. Meth. 2012, 9, 889–895, https://doi.org/10.1038/nmeth.2114.

146. Slaoui, M.; Fiette, L. Histopathology procedures: From tissue sampling to histopathological evaluation. In Drug Safety Evaluation: Methods and Protocols; Gautier, J.-C., Eds.; Humana Press: Totowa, NJ, USA, 2011; pp. 69–82, https://doi.org/10.1007/978-1-60761-849-2_4.

147. Yi, X.; Walia, E.; Babyn, P. Generative adversarial network in medical imaging: A review. Med. Image Anal. 2019, 58, 101552, https://doi.org/10.1016/j.media.2019.101552.

148. Rana, A.; Yauney, G.; Lowe, A.; Shah, P. Computational histological staining and destaining of prostate core biopsy RGB images with generative adversarial neural networks. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA); IEEE: Piscataway, NJ, USA, 2018; pp. 828–834, https://doi.org/10.1109/ICMLA.2018.00133.

149. Affonso, C.; Rossi, A.L.D.; Vieira, F.H.A.; de Leon Ferreira de Carvalho, A.C.P. Deep learning for biological image classification. Expert Syst. Appl. 2017, 85, 114–122, https://doi.org/10.1016/j.eswa.2017.05.039.

150. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.C.; Fei-Fei, L. ImageNet large scale visual recognition

challenge. Int. J. Comput. Vis. 2015, 115, 211–252, https://doi.org/10.1007/s11263-015-0816-y.

151. Yang, T.; Guo, X.; Wang, H.; Fu, S.; Wen, Y.; Yang, H. Magnetically optimized SERS assay for rapid detection of trace drug-related biomarkers in saliva and fingerprints. Biosens. Bioelectron. 2015, 68, 350–357, https://doi.org/10.1016/j.bios.2015.01.021.

152. Zhang, D.; Huang, L.; Liu, B.; Ni, H.; Sun, L.; Su, E.; Chen, H.; Gu, Z.; Zhao, X. Quantitative and ultrasensitive detection of multiplex cardiac biomarkers in lateral flow assay with core-shell SERS nanotags. Biosens. Bioelectron. 2018, 106, 204–211, https://doi.org/10.1016/j.bios.2018.01.062.

153. Schlücker, S. Surface-enhanced Raman spectroscopy: Concepts and chemical applications. Angew. Chem. Int. Ed. 2014, 53, 4756–4795, https://doi.org/10.1002/anie.201205748.

154. Chen, Y.; Zhang, Y.; Pan, F.; Liu, J.; Wang, K.; Zhang, C.; Cheng, S.; Lu, L.; Zhang, W.; Zhang, Z.; et al. Breath analysis based on surface-enhanced Raman scattering sensors distinguishes early and advanced gastric cancer patients from healthy persons. ACS Nano 2016, 10, 8169–8179, https://doi.org/10.1021/acsnano.6b01441.

155. Banaei, N.; Moshfegh, J.; Mohseni-Kabir, A.; Houghton, J.M.; Sun, Y.; Kim, B. Machine learning algorithms enhance the specificity of cancer biomarker detection using SERS-based immunoassays in microfluidic chips. RSC Adv. 2019, 9, 1859–1868, https://doi.org/10.1039/C8RA08930B.

156. Guselnikova, O.; Postnikov, P.; Pershina, A.; Svorcik, V.; Lyutakov, O. Express and portable label-free DNA detection and recognition with SERS platform based on functional Au grating. Appl. Surf. Sci. 2019, 470, 219–227, https://doi.org/10.1016/j.apsusc.2018.11.092.

157. Goodacre, R.; Graham, D.; Faulds, K. Recent developments in quantitative SERS: Moving towards absolute quantification. TrAC—Trends Anal. Chem. 2018, 102, 359–368, https://doi.org/10.1016/j.trac.2018.03.005.

158. Guselnikova, O.; Trelin, A.; Skvortsova, A.; Ulbrich, P.; Postnikov, P.; Pershina, A.; Sykora, D.; Svorcik, V.; Lyutakov, O. Label-free surface-enhanced Raman spectroscopy with artificial neural network technique for recognition photoinduced DNA damage. Biosens. Bioelectron. 2019, 145, 111718, https://doi.org/10.1016/j.bios.2019.111718.

159. Thrift, W.J.; Cabuslay, A.; Laird, A.B.; Ranjbar, S.; Hochbaum, A.I.; Ragan, R. Surface-enhanced Raman scattering-based odor compass: Locating multiple chemical sources and pathogens. ACS Sens. 2019, 4, 2311–2319, https://doi.org/10.1021/acssensors.9b00809.

160. Thrift, W.J.; Nguyen, C.Q.; Wang, J.; Kahn, J.E.; Dong, R.; Laird, A.B.; Ragan, R. Improved regressions with convolutional neural networks for surface enhanced Raman scattering sensing of metabolite biomarkers. Proc. SPIE 2019, 11089, 1108907, https://doi.org/10.1117/12.2535410.

161. Thrift, W.J.; Ragan, R. Quantification of analyte concentration in the single molecule regime using convolutional neural networks. Anal. Chem. 2019, 91, 13337–13342, https://doi.org/10.1021/acs.analchem.9b03599.

162. Cheng, L.; Meng, Q.H.; Lilienthal, A.J.; Qi, P.F. Development of compact electronic noses: A review. Meas. Sci. Technol. 2021, 32, 062002, https://doi.org/10.1088/1361-6501/abef3b.

163. Jiang, L.; Hassan, M.M.; Ali, S.; Li, H.; Sheng, R.; Chen, Q. Evolving trends in SERS-based techniques for food quality and safety: A review. Trends Food Sci. Technol. 2021, 112, 225–240, https://doi.org/10.1016/j.tifs.2021.04.006.

164. Pang, S.; Yang, T.; He, L. Review of surface enhanced Raman spectroscopic (SERS) detection of synthetic chemical pesticides. TrAC—Trends Anal. Chem. 2016, 85, 73–82, https://doi.org/10.1016/j.trac.2016.06.017.

165. Barreiros dos Santos, M.; Queirós, R.B.; Geraldes, Á.; Marques, C.; Vilas-Boas, V.; Dieguez, L.; Paz, E.; Ferreira, R.; Morais, J.; Vasconcelos, V.; et al. Portable sensing system based on electrochemical impedance spectroscopy for the simultaneous quantification of free and total microcystin-LR in freshwaters. Biosens. Bioelectron. 2019, 142, 111550, https://doi.org/10.1016/j.bios.2019.111550.

166. Huang, X.; Li, Y.; Xu, X.; Wang, R.; Yao, J.; Han, W.; Wei, M.; Chen, J.; Xuan, W.; Sun, L. High-precision lensless microscope on a chip based on in-line holographic imaging. Sensors 2021, 21, 720, https://doi.org/10.3390/s21030720.

167. Zhang, R.; Belwal, T.; Li, L.; Lin, X.; Xu, Y.; Luo, Z. Nanomaterial-based biosensors for sensing key foodborne pathogens: Advances from recent decades. Compr. Rev. Food Sci. Food Saf. 2020, 19, 1465–1487, https://doi.org/10.1111/1541-4337.12576.

168. Wang, Q.; Wei, H.; Zhang, Z.; Wang, E.; Dong, S. Nanozyme: An emerging alternative to natural enzyme for biosensing and immunoassay. TrAC—Trends Anal. Chem. 2018, 105, 218–224, https://doi.org/10.1016/j.trac.2018.05.012.

169. Zhu, X.; Liu, P.; Ge, Y.; Wu, R.; Xue, T.; Sheng, Y.; Ai, S.; Tang, K.; Wen, Y. MoS2/MWCNTs porous nanohybrid network with oxidase-like characteristic as electrochemical nanozyme sensor coupled with machine learning for intelligent analysis of carbendazim. J. Electroanal. Chem. 2020, 862, 113940, https://doi.org/10.1016/j.jelechem.2020.113940.

170. Mahmudunnabi, G.R.; Farhana, F.Z.; Kashaninejad, N.; Firoz, S.H.; Shim, Y.-B.; Shiddiky, M.J.A. Nanozyme-based electrochemical biosensors for disease biomarker detection. Analyst 2020, 145, 4398–4420, https://doi.org/10.1039/D0AN00558D.

171. Zhang, X.; Wu, D.; Zhou, X.; Yu, Y.; Liu, J.; Hu, N.; Wang, H.; Li, G.; Wu, Y. Recent progress in the construction of nanozyme-based biosensors and their applications to food safety assay. TrAC—Trends Anal. Chem. 2019, 121, 115668, https://doi.org/10.1016/j.trac.2019.115668.

172. Riley, P. Three pitfalls to avoid in machine learning. Nature 2019, 572, 27–29, https://doi.org/10.1038/d41586-019-02307-y.

173. Stodden, V.; McNutt, M.; Bailey, D.H.; Deelman, E.; Gil, Y.; Hanson, B.; Heroux, M.A.; Ioannidis, J.P.A.; Taufer, M. Enhancing reproducibility for computational methods. Science 2016, 354, 1240–1241, https://doi.org/10.1126/science.aah6168.

APPENDIX B

Evaluation of Computational Phage Detection Tools for Metagenomic Datasets

Kenneth E. Schackart III[1], Jessica B. Graham[2], and Alise J. Ponsero[1,3,4,*], Bonnie L. Hurwitz[1,3]

[1]Department of Biosystems Engineering, The University of Arizona, Tucson, Arizona 85721, USA

[2]Department of Biomedical Engineering, The University of Arizona, Tucson, Arizona 85721, USA

[3]BIO5 Institute, The University of Arizona, Tucson, Arizona 85721, USA

[4]Human Microbiome Research Program, Faculty of Medicine, University of Helsinki, Helsinki, Finland

*Correspondence and requests for materials should be addressed to Alise Ponsero (aponsero@arizona.edu)

Submitted to Frontiers in Microbiology, 24 October 2022

Abstract

New tools for detecting phage in metagenomes are being rapidly developed. A critical need has emerged to develop systematic benchmarks for these tools. We collected 19 metagenomic phage detection tools based on a literature survey, 9 of which could be installed and run at scale. Those 9 tools were assessed on several benchmark challenges. Fragmented reference genomes are used to assess the effects of fragment length, low viral content, phage taxonomy, robustness to eukaryotic contamination, and computational resource usage. Simulated metagenomes are used to assess the effects of sequencing and assembly quality on the tool performances. Finally, real human gut metagenomes are used to assess the differences and similarities in the phage communities predicted by the tools. We find that the various tools yield strikingly different results. Generally, tools that use a homology approach (VirSorter, MARVEL, viralVerify, VIBRANT, and VirSorter2) demonstrate low false positive rates and robustness to eukaryotic contamination. Conversely, tools that use a sequence composition approach (VirFinder, DeepVirFinder, Seeker), and MetaPhinder, have higher sensitivity, including to phages with less representation in reference databases. These differences led to widely differing predicted phage communities in human gut metagenomes, with nearly 80% of contigs being marked as phage by at least one tool and a maximum overlap of 38.8% between any two tools. Importantly, the benchmark datasets developed in this study are publicly available and reusable to enable the future comparability of new tools developed.

Keywords:

Metagenomics, bacteriophage, virome, benchmark

1. Introduction

Prokaryotic viruses called bacteriophages (phages) are the most abundant biological entity in most ecosystems (Ofir and Sorek 2018) and profoundly impact the ecology of natural ecosystems (Breitbart and Rohwer 2005; Blazanin and Turner 2021). For example, marine viruses have massive effects on ocean biochemistry, influencing nutrient cycling and carbon sequestration by altering host-driven processes through controlling bacterial population growth and altering metabolic function, (as reviewed in Fuhrman 1999; Breitbart et al. 2018; Hurwitz and U'Ren 2016). Additionally, recent studies have demonstrated the importance of phages in shaping the human microbiota and interacting with human health (as reviewed in Y. Li, Handley, and Baldridge 2021; Manrique, Dills, and Young 2017; Edlund et al. 2015; Sharma et al. 2018). Next-generation sequencing techniques enable the identification of an exponential number of novel phages but also allow for a better understanding of phage populations in multiple ecosystems (Breitbart et al. 2002).

Despite the increasing number of virome studies, identifying viral sequences in metagenomic datasets is still computationally challenging. Since viruses lack a universal gene marker (*e.g.,* 16S rRNA in prokaryotes), earlier bioinformatics methods to identify viruses from metagenomes often relied on sequence alignment methods against reference genome databases. Strikingly, in gut viromes, 75% to 99% of viral reads do not produce significant alignments to any known viral genome. This large range in alignable sequences can be partially explained by the high diversity, fast phage evolution, and their ability to integrate into their host genome and be mistaken as bacterial. Indeed, an inherent limitation to genome comparison approaches is the database completeness and clear separation between host and viral DNA. Moreover, these reference-based methods typically cannot identify novel phage sequences. To address these

limitations, several dedicated computational tools and approaches were proposed. In 2015, the tool VirSorter (Roux et al. 2015) was released, identifying phage sequences by enrichment of viral hallmark genes, depletion of cellular genes indicated by reduced Pfam hits, and strand shifts. In 2016, MetaPhinder took into account the mosaicism of phage sequences by integrating hits to multiple genomes to classify a sequence as host or viral (Jurtz et al. 2016).

Recently, bioinformatic tools leverage machine learning algorithms to identify features of viral origin, and typically allow for a broader recall of previously unknown sequences than alignment-based approaches. Chosen features are genes and gene density (Amgarten et al. 2018; Guo et al. 2021; Kieft, Zhou, and Anantharaman 2020; Antipov et al. 2020; Tisza et al. 2021) and protein families (Amgarten et al. 2018) that are used to train classification models including random forest (Amgarten et al. 2018; Guo et al. 2021), naive Bayes (Antipov et al. 2020), and neural network (Kieft, Zhou, and Anantharaman 2020). Interestingly, some authors also proposed using the differential k-mer (short sequences of length k) frequencies between phages and prokaryotes for sequence classification (Ren et al. 2017; Deaton, Yu, and Quake 2017). These methods allow the detection of shorter phage sequences, as they do not require multiple open reading frames (ORF) for classification that are difficult to obtain in fragmentary metagenomic data. However, the classification results and the rationale behind the classification are typically difficult to interpret.

All in all, between 2015 and 2021, we identified 19 published tools designed for detecting phage in metagenomes, making the development of benchmarking datasets critical for exploring the limitations and biases of the currently available tools but also facilitating future tool development. Similar benchmark efforts are currently available for other computational tasks such as metagenome assembly, binning, and taxonomic profiling (Sczyrba et al. 2017; Meyer et al.

2019). Recently, several efforts to benchmark phage detection tools have been published, and explore the ability of these tools to correctly identify and classify dsDNA viruses and curate auxiliary metabolic genes (Ho et al. 2022; Pratama et al. 2021). However, the potential impact of parameters such as the sequence length, sequencing error, eukaryotic contamination, quality of assembly, and phage taxonomy on the tool's classification performance is not explored.

In this study, we developed a series of benchmark datasets, each aiming at assessing a precise classification challenge in detecting phage in metagenomic datasets and evaluated nine phage metagenomic detection tools published before July 2021. However, this work does not include tools specifically intended to detect integrated prophage in complete bacterial genomes, for which prior benchmarking efforts are already available (Roach et al. 2022). Importantly, we ensured the availability and reusability of the developed benchmark datasets and described how researchers could utilize them for benchmarking new phage detection tools (doi.org/10.5281/zenodo.7194617).

2. Methods

2.1. Phage Detection Tools

Tools were categorized into two broad groups. Homology-based tools are those that utilize a reference database at the time of classification to search for homologues. Sequence-based tools are those that classify using a model trained on sequence features such as k-mer frequencies.

Each of the evaluated tools included in this study was installed from the recommended source following the authors' instructions. When tools were available from several sources, Bioconda was preferred due to simplified dependency management. Tools that could not be obtained through Bioconda were directly cloned from GitHub or Sourceforge. Table 1 summarizes the tool and version number when available, the category, classification method, and how tools

were obtained. Tools with "-" under distribution were not used for further benchmarking, and reasons for doing so are presented in Results.

Table B - 1

Overview of metagenomic phage detection tools

Tool & version	Category	Method	Distribution	Reference
DeepVirFinder (1.0)	Sequence	*k*-mer based deep learning neural net	GitHub	(Ren et al. 2020)
MARVEL (0.2)	Homology	Random forest utilizing gene density, strand shifts, and proteins.	GitHub	(Amgarten et al. 2018)
MetaPhinder	Homology	Integrated analysis of BLASTn hits to a phage database	GitHub	(Jurtz et al. 2016)
PhaMers	Sequence	*k*-Nearest Neighbors and centroid proximity metric of *k*mers	-	(Deaton, Yu, and Quake 2017)
PPR-Meta	Sequence	Convolutional Neural Network (CNN) of one-hot encodings of nucleobases and codons	-	(Fang et al. 2019)
RNN-VirSeeker	Sequence	Long Short-Term Memory (LSTM) of sequences	-	(F. Liu et al. 2020)
Seeker	Sequence	LSTM of sequences	PyPi	(Auslander et al. 2020)
Unlimited Breadsticks	Homology	HMM of virus hallmark genes	GitHub	(Tisza et al. 2021)
VIBRANT (1.0.1)	Homology	Neural network of protein signatures including ratios of KEGG, VOG, and PFAM hits, and presence of key viral-like genes.	Bioconda	(Kieft, Zhou, and Anantharaman 2020)

viralVerify (1.1)	Homology	Naive Bayes classifier using an hmmsearch of genes predicted with Prodigal	Bioconda	(Antipov et al. 2020)
ViraMiner	Sequence	CNN of one-hot encoded nucleobases	-	(Tampuu et al. 2019)
VirFinder (1.1)	Sequence	Logistic regression using *k*-mers	Bioconda	(Ren et al. 2017)
VirMine	Homology	BLAST search of ORFs against viral and non-viral databases	-	(Garretto, Hatzopoulos, and Putonti 2019)
VirMiner	Homology	Random forest (RF) based on functional profiling and protein homology	-	(Zheng et al. 2019)
VirNet	Sequence	Deep attention model of sequences	-	(Abdelkareem et al. 2018)
VIROME	Homology	Functional and taxonomic information based on ORF homology	-	(Wommack et al. 2012)
VirSorter	Homology	Gene homology including enrichment of viral-like and short genes, and depletion of PFAM hits and strand shifts	wget	(Roux et al. 2015)
VirSorter2 (2.2)	Homology	Random forest classifiers using an hmmsearch of genes predicted with Prodigal	Bioconda	(Guo et al. 2021)
VirusSeeker	Homology	BLAST search against virus database, followed by search against full NCBI database to remove false positives	-	(G. Zhao et al. 2017)

2.2. Datasets

This study leverages 4 benchmark datasets: (1) *genome fragment set* used for assessing the effect of contig length, low viral abundance, eukaryotic contamination, and potential bias toward

certain groups of phages, (2) *simulated phageome set* used to explore the effect of sequencing error on the classification, (3) *simulated metagenome set*, used for exploring the effect of the quality of assembly, and viral abundance in samples. Finally, the study includes a real gut metagenome dataset from colorectal cancer (CRC) patients compared to healthy controls: (4) *CRC dataset* used to compare the results of the tools on a real metagenomic dataset.

2.2.1 Set 1: Genome Fragment Set

All complete bacterial, fungal, and viral genomes were downloaded from the RefSeq database on 14 June 2021 (O'Leary et al. 2016). These genomes were fragmented into non-overlapping adjacent fragments of lengths 500, 1000, 3000, and 5000 nucleotides. In total, 379 archaeal, 21,788 bacterial, 18 fungal, and 11,156 viral genomes were obtained and fragmented, of which 1,483 were phage. From those fragmented genomes, 10,000 fragments were randomly selected from each length and each superkingdom. The resulting set includes four subsets: 500, 1000, 3000, and 5000 bp, each with 10k fragments from the four superkingdoms for a total of 40k fragments per length subset. This collection of unmodified fragmented reference genomes is referred to as the *genome fragment set*.

2.2.2. Set 2: Simulated Phageome Set

InSilicoSeq v 1.5.4 (Gourlé et al. 2019) was used for creating simulated reads from phage genomes. This tool creates an error model of per-base quality (Phred) scores using Kernel Density Estimation, trained on real sequencing reads. InSilicoSeq was chosen due to its computational efficiency (McElroy, Luciani, and Thomas 2012), simplicity of use and documentation (Yu et al. 2020), and ability to simulate Illumina sequencing instead of 454 technology (Richter et al. 2008; M. Zhao, Liu, and Qu 2017). Additionally, this tool has been demonstrated to generate reads with

realistic quality score distributions for several sequencing platforms, including MiSeq, HiSeq, and NovaSeq (Yu et al. 2020; Gourlé et al. 2019).

Three "phageome" profiles were created by randomly selecting 500 phage genomes per profile from the downloaded RefSeq database. Reads were simulated using InSilicoSeq, specifying 30x coverage of all genomes. Reads simulated using each of the three built-in error models were created for each profile. Simulated reads from InSilicoSeq were assembled with MEGAHIT v1.2.9 (D. Li et al. 2015) and binned with MetaBAT 2 v2:2.15 and using Bowtie2 v2.4.5 for indexing (Kang et al. 2019; Langmead and Salzberg 2012).

To determine the genomic origin of each contig, BLAST v2.12.0+ was used for alignment. Since the genomes used for read simulation for each profile were known, a local BLAST database was created using those same genomes for each simulated metagenome, reducing spurious hits. Alignment was done using the MEGABLAST mode of BLASTn, with an e-value of 1e-20. Even with the limited BLAST databases, it was common for contigs to have significant hits to several genomes. To determine the "true" origin of the contigs, a basic decision tree was used which is shown in Supplemental Fig B-S3.

In total, the *simulated phageome set* is comprised of nine phageome assemblies and bin sets (3 profiles * 3 error models).

2.2.3. Set 3: Simulated Metagenome Set

The same simulation and binning steps used in the Set 2 were used to generate a set of simulated metagenomes. Five marine samples were used as the basis of this dataset. Three were from the Hawaii Ocean Time-series (HOT) program (SRR5720259, SRR5720320, SRR6507280) (Karl and Lukas 1996), and two were from the Amazon continuum dataset (SRR4831655, SRR4831664) (Satinsky et al. 2014). Raw sequencing data were downloaded from Sequence Read

Archive (SRA), using fastqc v0.11.9 and trimGalore v0.6.6 (Andrews and others 2010; "Babraham Bioinformatics - Trim Galore!" n.d.). Briefly, reads with average base quality score below 20 were removed, and those with adapters and poly-G sequences were trimmed. After trimming, reads with a length < 20bp were filtered out. After quality control (QC), taxonomic abundance profiles of the bacterial and phage population in each sample were obtained using Kraken2 (Wood, Lu, and Langmead 2019) and Bracken (Lu et al. 2017) against the PlusPF database (version 5/17/2021 available at https://benlangmead.github.io/aws-indexes/k2). The abundance profiles were used as input for InSilicoSeq, using reference genomes obtained from the RefSeq database. Additionally, for any profile with a phage abundance below 5% of reads, the profile was supplemented with additional phages by adding a minimum of 10 phages known to infect the top non-viral organisms in the profile. 20M Simulated reads for each profile were generated using the three built-in error models (HiSeq, MiSeq, and NovaSeq). The 15 resulting assemblies and bins are referred to as the *simulated metagenomes set*.

2.2.4. Set 4: CRC Dataset

We also included a real-metagenomic dataset from a published study that used fecal shotgun metagenomics to characterize stool microbial populations from colorectal cancer (CRC) patients compared to healthy controls with a total of 198 samples (https://doi.org/10.15252/msb.20145645). Raw sequencing data were downloaded from SRA (PRJEB6070) and were quality filtered using fastqc v0.11.9 and trimGalore v0.6.6. Briefly, reads with an average base pair quality score below 20 were removed, and adapters and poly-G sequences were trimmed. After trimming, reads with a length < 20bp were filtered out. Quality-filtered sequences were screened to remove human sequences using bowtie2 v2.4.2 against a non-

redundant version of the Genome Reference Consortium Human Build 38, patch release 7 (available at PRJNA31257 in NCBI).

After QC and human read filtering, the reads were assembled using Megahit v1.2.9. The code of the pipeline used for the assembly is available on Github (https://github.com/aponsero/Assembly_metagenomes). Megahit was run on the paired-end reads or single-end reads using the default parameters (referred to as the simple assembly). Additionally, a co-assembly of the multiple runs per BioSample was also performed (referred to as the co-assembly). Assemblies were binned with MetaBAT 2 v2:2.15 and using Bowtie2 v2.4.5 for indexing.

CheckV v1.0.1 was run on all assemblies to assess viral and bacterial gene content. This would be used to roughly assess the quality of the predictions made by the tools.

2.3. Classification of The Datasets

Snakemake was used as a workflow manager for running the tools (Köster and Rahmann 2012). This pipeline was implemented on the Puma High-Performance Compute (HPC) cluster at the University of Arizona using SLURM (Yoo, Jette, and Grondona 2003). While running the tools, the following metrics were collected by Snakemake: runtime and CPU time, peak memory usage, and file write operations.

When running the tools, the default parameters, modes, and databases were used to replicate those intended for use by the authors. DeepVirFinder was run without a length cutoff. MetaPhinder was run using the default database. Seeker was run using the command-line executable binary instead of the Python package. VIBRANT was run in standard (not virome) mode, with the default minimum length (1000 bp) and number of ORFs (4). viralVerify utilized the default database. Virsorter was run using the default (RefSeq) database, in non-virome mode,

and BLASTP as the default was used instead of DIAMOND. Virsorter2 was run to identify only dsDNAphage and ssDNAphage, allowing for proviruses by not using the "–no-pro-virus" flag and not limiting the number of ORFs.

All of the tools classified the fragments in the *genome fragment set* except for MARVEL, which requires bins as input. VIBRANT and VirSorter do not classify fragments shorter than 1000 nucleotides (nt), so there is no data for these tools for the 500 nt fragments. Default parameters were used for cutoff thresholds when the option was provided. To simplify the comparison of the classification performances, all predictions were binned into "phage" or "non-phage" classes. VirFinder and DeepVirFinder return a prediction score, which were used to bin fragments at a 0.5 cutoff. VIBRANT predicts both prophage and lytic virus labels, both of which were considered to be classified as phage. VirSorter also predicts prophage and lytic labels, and assigns a confidence category, all of which were considered to be classified as phage.

The simulated datasets (*simulated phageome set* and *simulated metagenome set*), and the *CRC dataset* were classified by all tools. Assembled reads were classified by all tools except MARVEL. MARVEL was given binned assemblies for classification. Resource usage required for binning was included in the resource usage benchmarking for MARVEL.

2.4. Performance Assessment

Several performance metrics are assessed for each of the challenge datasets. These are precision, sensitivity (recall), specificity, *F*1, and AUPRC, as defined below. In these definitions, a "positive" is a phage sequence, while a negative is anything that is not phage. Accordingly, a true positive (TP) is a phage sequence that has been correctly labeled as phage, a True Negative (TN) is a non-phage sequence correctly labeled as such, a False Positive (FP) is a non-phage sequence labeled as phage, and a False Negative (FN) is a phage sequence not labeled as phage.

Precision is the portion of all predicted phage sequences that are indeed phage (eq. 1). Sensitivity, also known as recall, is the proportion of all true phage sequences that were correctly identified (eq. 2). Specificity is the proportion of all non-phage sequences that were correctly labeled (eq. 3). *F*1 score is the harmonic mean of precision and sensitivity (eq. 4). The area under the precision recall curve (AUPRC) is a measure of precision over the sensitivity range, given a varying classification threshold for a continuous predictive output value (eq. 5).

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$sensitivity\ (recall) = \frac{TP}{TP + FN} \tag{2}$$

$$specificity = \frac{TN}{TN + FP} \tag{3}$$

$$F1 = \frac{2 \times precision \times sensitivity}{precision + sensitivity} \tag{4}$$

$$AUPRC = \int_0^1 p\ dr\ ; p = precision, r = recall \tag{5}$$

3. Results

3.1. Installation of Tools

A total of 19 tools were collected based on a survey of the literature as of July 2021. However, several tools were omitted from further investigation for the following reasons: (1) the creation of runtime exceptions (PhaMers and VirMine), (2) tools with hard-coded paths that require the user to modify source code (RNN-VirSeeker and VirusSeeker), (3) lack of clear installation instructions and documentation (ViraMiner and VirNet), (4) tools that are unscalable due to web server usage (VirMiner and VIROME), and (5) inability to run instances of the tool on different cores from the same directory (PPR-Meta). Finally, Cenote Unlimited Breadsticks could

be installed and run but did not classify any of the fragments as viral, and was excluded from the benchmark analysis, but included in the resource usage comparison.

3.2. Resource Usage

Computational resource usage was benchmarked using the *genome fragment set* since the quantity and length of genomic fragments were known and balanced. Pre- and post-processing steps were excluded from these measurements. For tools that did not allow the user to specify the output directory (MARVEL and Seeker), we also included time to move output files to the correct output directory.

The total time (in CPU time) for each tool included: (1) CPU time to run the tool summed for user and system and (2) the amount of time to read and write data while classifying the *genome fragment set* (Fig B-1). For some tools, CPU time was highly variable (MetaPhinder, VIBRANT, viralVerify, and to a lesser extent for VirSorter, VirSorter2, and Seeker). Seeker generally had the longest CPU times, even for shorter fragments. While DeepVirFinder was consistently fast, its real-world performance was hindered due to its use of the Theano backend. While multiple jobs can be submitted in parallel, the Theano backend can only process one dataset at a time for serial processing. This led to long-running jobs for DeepVirFinder, but deceptively low CPU time measures.

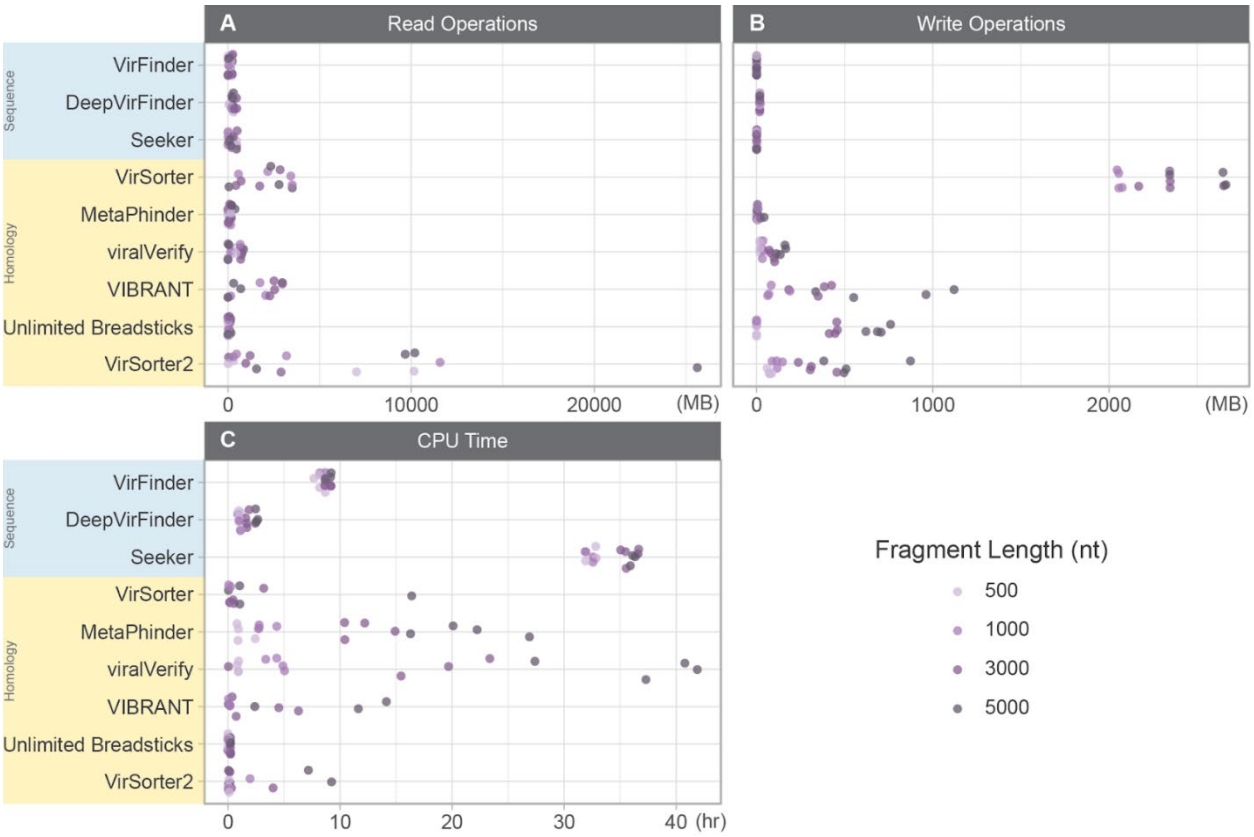

Figure B - 1

Resource usage while classifying 10k genome fragments of various lengths (500, 1000, 3000, and 5000 nt). A) Read operations (MB) B) Write operations (MB) C) CPU time (hr) summed for user and system. Sequence-based tools are in blue, homology-based tools are in yellow.

3.3. Benchmark Challenge 1: Classification of Genome Fragments

3.3.1. Effect of Contig Length

We first evaluated the effect of contig length on each tool's performance. To assess this effect, the *genome fragment set (Set 1)* was used as input. Fig B-2A shows *F*1 score for increasing fragment lengths (500, 1000, 3000, and 5000 nt). As expected, homology-based tools such as VIBRANT, ViralVerify, and VirSorter2 were strongly affected by fragment length, with performance increasing with length. VirSorter had the lowest *F*1 score for all lengths and had only

a marginal increase in *F*1 with increasing length. The sequence-based tools (DeepVirFinder, VirFinder, and Seeker), as well as MetaPhinder, were largely unaffected by fragment length.

While *F*1 score illustrates overall changes in classification performance due to fragment length, each tool's performance is affected in different ways (Fig B-2B). VIBRANT, viralVerify, and VirSorter2 demonstrate fairly consistent and high precision but have length-dependent sensitivity, whereas DeepVirFinder, MetaPhinder, Seeker, and VirFinder demonstrate fairly consistent sensitivity and precision. Generally, length-dependent sensitivity is a property of homology-based tools where longer fragments are needed for classification. Interestingly, MetaPhinder (a homology-based tool) exhibits a pattern similar to the sequence-based tools for this property.

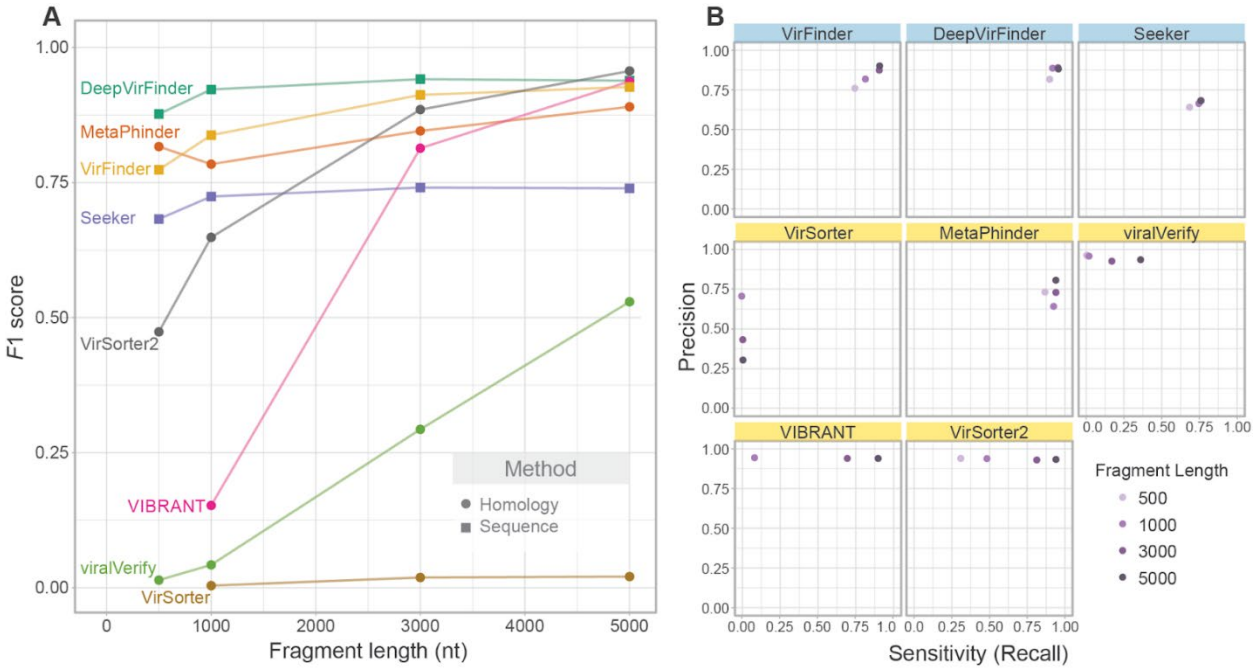

Figure B - 2

Effect of fragment length on classification performance. Only phage and bacterial sequence fragments are included. A) Balanced *F*1 score plotted against fragment length (nt) and B) Balanced

precision plotted against sensitivity for four fragment lengths. Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

Several tools output a continuous classification score metric. For these tools (VirFinder, DeepVirFinder, Seeker, MetaPhinder, and viralVerify), the threshold used will affect precision and sensitivity. Although the default threshold was used, the effect of this threshold can be seen in the precision-recall curves (Supplementary Fig B-S1) and AUPRC (Fig B-S2). These tools all showed lower AUPRC for shorter contigs, with DeepVirFinder outperforming the other tools even on shorter contigs.

3.3.2. Low Viral Content

In the above section, precision is computed based on a balanced dataset (equal quantities of phage and bacteria). However, this gives a highly optimistic estimate of precision. For a given false positive rate (FPR), precision will drop significantly when phage content is low. To illustrate this, the FPR (eq. 6) was taken from the classification of the *genome fragment set*, and precision was extrapolated to hypothetical community compositions ranging from 0% to 100% non-viral fragments (Fig B-3). 50% represents a balanced dataset.

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

For communities with low viral content, precision decreases for nearly all tools. Notably, viralVerify did not falsely classify any 500 nt fragments as phage, thus its hypothetical precision remains perfect for that fragment length. VIBRANT, viralVerify, and VirSorter2 maintain fairly high precision but still drop below 0.5 for communities with low viral content.

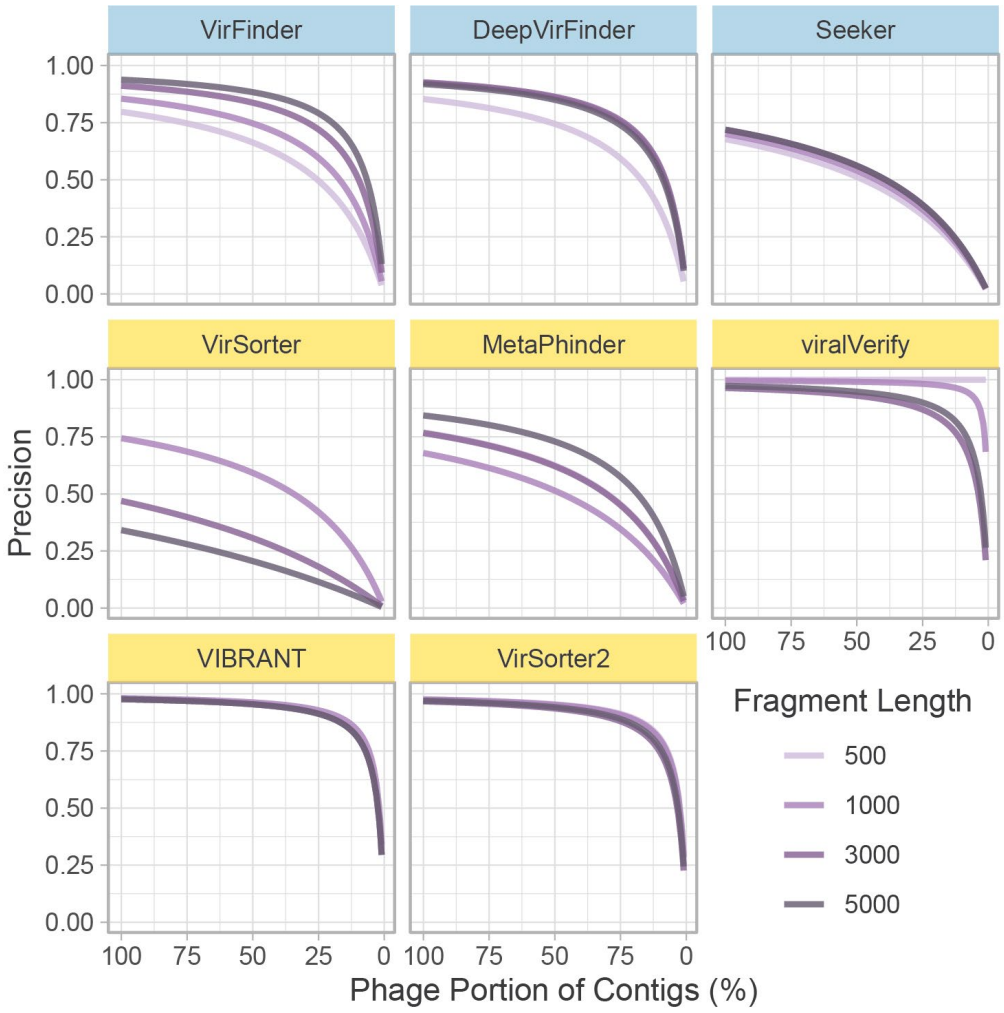

Figure B - 3

Extrapolated precision calculated from FPR of each tool at four fragment lengths (500, 1000, 3000, 5000 nt). Precision is calculated for communities composed of varying levels of non-viral fragments from 0% (all phage) to 100% (all non-phage). Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

3.3.3. Effect of Phage Taxonomy

The lack of phage diversity and bias for certain phage groups in reference databases leads to challenges in training models and propensity for tools to retrieve fewer phages from less represented phage groups. The majority (ca. 93%) of phages in RefSeq belong to the order

Caudovirales (recently renamed as the class Caudoviricetes; Turner, Kropinski, and Adriaenssens 2021). Importantly, all tools were shown to have reduced sensitivity for non-caudovirales sequences. In particular homology-based tools showed a drastically reduced sensitivity toward these phages (Fig B-4).

While the limitation of database composition is mitigated by sequence-based compared to homology-based approaches, a slightly lower sensitivity for non-caudoviral phages was nonetheless observed.

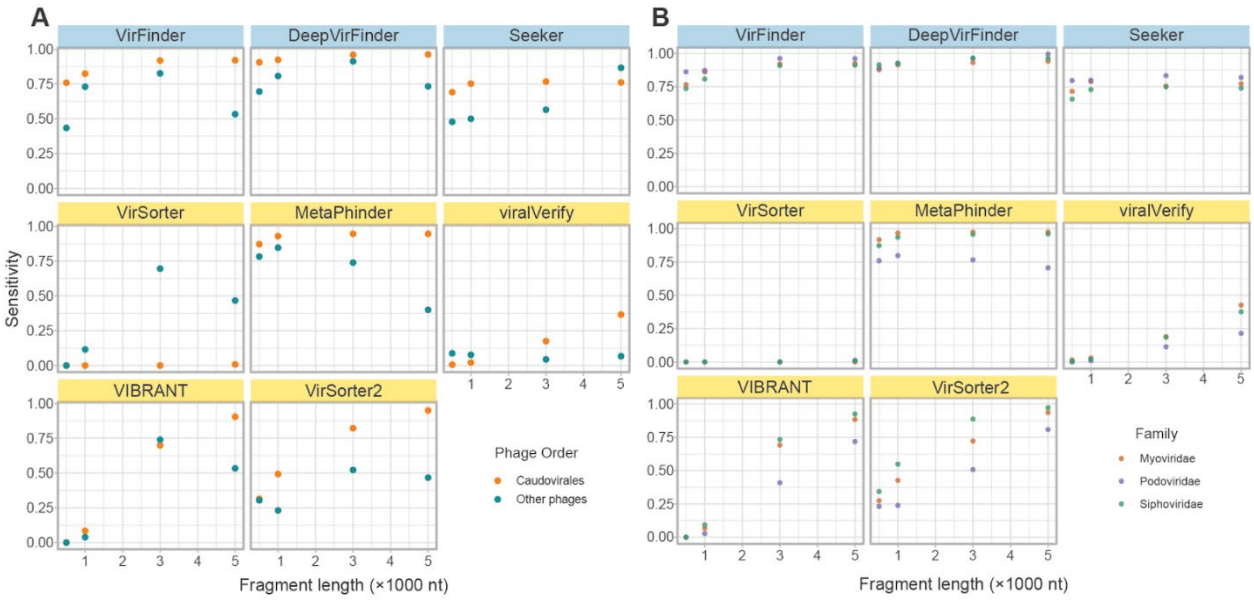

Figure B - 4

Effects of phage taxonomy on sensitivity. A) Sensitivity plotted against fragment length, comparing bacteriophages in the order Caudovirales and those in other orders. B) Sensitivity plotted against fragment length, comparing bacteriophages in the top three families of the order *Caudovirales*. Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

Even within the caudoviral order, the three main families are detected unequally by the tools. *Siphoviridae* constitutes the greatest phage family represented in RefSeq (69.9%), followed by *Myoviridae* (15.6%) and *Podoviridae* (7.28%). This is apparent in the *fragmented genomes set* due to random sampling from the fragmented genomes; of the caudoviruses, more fragments came from *Siphoviridae* and *Myoviridae* than *Podoviridae*. Fig B-4 demonstrates how sensitivity is decreased for the retrieval of *Siphoviridae* and *Podoviridae* sequences compared to *Myoviridae* in particular MetaPhinder, VIBRANT, and VirSorter2.

3.3.4. Eukaryotic Contamination

A concern for sequence-based tools is specificity when faced with eukaryotic contamination, due to the lack of eukaryotic sequences in the training sets (Ponsero and Hurwitz 2019). As part of the *genome fragment set*, the tools classified 10k eukaryotic genome fragments of 4 lengths from fungi in the phyla Ascomycota and Basidiomycota. The specificity of these eukaryotic fragments is shown in Fig B-5. All homology-based tools are extremely robust to eukaryotic contamination even for short fragments. However, sequence-based tools and Metaphinder show much lower specificity, frequently misclassifying eukaryotic fragments as viral, with an FPR around 0.5. Notably, Seeker shows a sensitivity that is worse than a random chance binary classification, classifying nearly all eukaryotic contigs as viral.

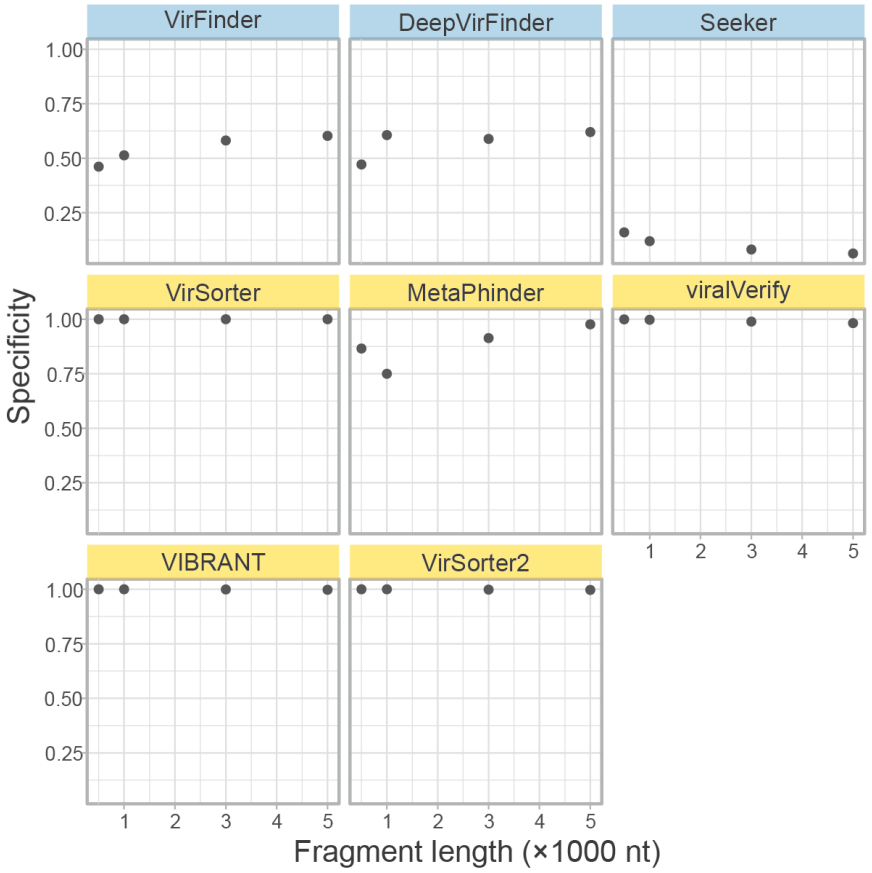

Figure B - 5

Classification specificity of eukaryotic genome fragments (n=10k). Eukaryotic fragments are from the Ascomycota and Basidiomycota phyla. Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

3.4. Benchmark Challenge 2: Classification of Simulated Metagenomic Sequences

To compare the relative performance of the tools when faced with read errors from different sequencing technologies and potential assembly error, the *simulated phageome set* and *simulated metagenome set* were created from simulated reads generated using error models that represent 3 popular sequencing platforms: HiSeq, MiSeq, and NovaSeq. Each technology has a unique per-base error rate, as modeled by InSilicoSeq, as well as differing read lengths. We aimed to assess how these differences affect assembly and tool performance.

3.4.1. Simulated Phageomes

To directly assess the effect of sequencing error and assembly on classification sensitivity, the *simulated phageome set* was classified by each tool. In this dataset, simulated reads were obtained from phage genomes and assembled into contigs. The assembled contigs' length varied from 500 to 309,196 bp, with a median length of 949 bp. Unlike in the *genome fragment set*, simulated contigs could be used to assess MARVEL, which requires binned sequences for classification. Each tool's sensitivity was calculated using the assembled contigs grouped by length (Fig B-6).

The results are largely consistent with those obtained using the *genome fragment set*. DeepVirFinder and MetaPhinder, followed by VirFinder and Seeker, show the highest and most consistent sensitivity across all contig lengths. VIBRANT, VirSorter, and VirSorter2 performed well for longer contigs, but sensitivity suffers as contig length decreases. VirSorter's sensitivity begins to improve at about 10^4 bp and increases greatly for $10^{4.5}$ and 10^5 bp. MARVEL, while better than VirSorter on short contigs, demonstrates the lowest sensitivity for long contigs. Importantly, MARVEL shows more variability in sensitivity for a given length. Indeed, the tool performs classification of contigs after binning, and since contigs of various lengths may be present in the same bin, we observe that the tool's performance is less tightly coupled to contig length.

The three error models produce reads of different lengths (HiSeq 125 bp, NovaSeq 150 bp, and MiSeq 300 bp). This led to a significant difference in contig lengths between MiSeq and the other two error models (based on Wilcoxon rank sum test with Bonferroni adjusted p-value, $p < 0.05$). However, the sensitivity at a given length was similar across error models, suggesting that the difference in sequencing technologies is erased by the assembly process.

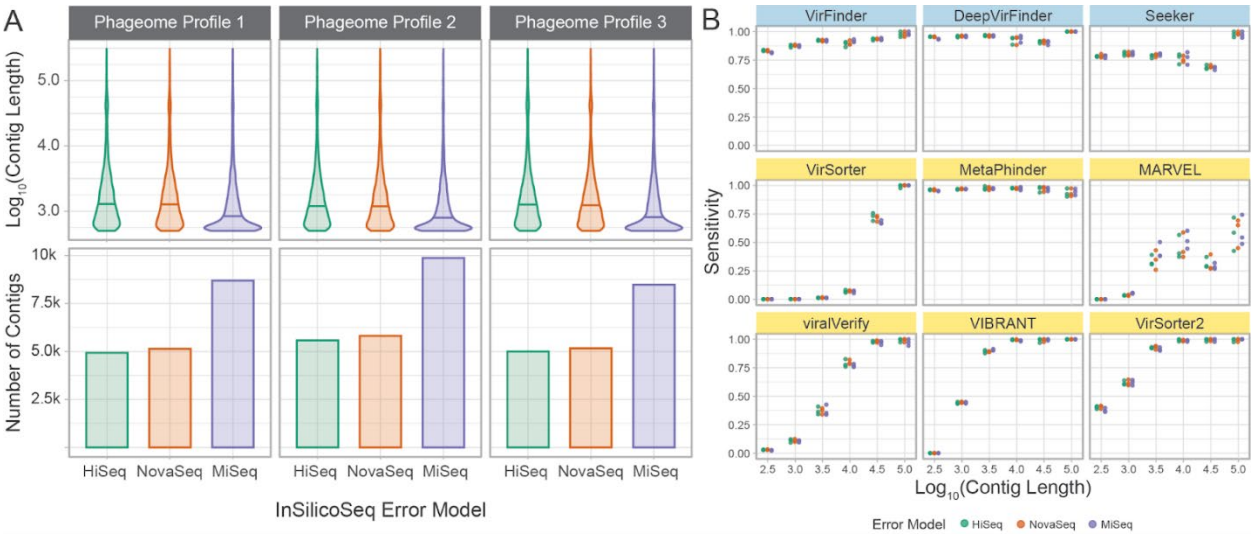

Figure B - 6

Simulated phageome results. A) Length distributions, and number of, assembled contigs constituting the nine phageomes generated using the three phageome profiles and three error models. B) Sensitivity of classifying simulated phage contigs. Contigs were grouped by length (x-axis) for computation of sensitivity. Error models are ordered by increasing read length. Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

3.4.2. Simulated Metagenomes

The *simulated metagenome set* was produced from the bacterial and phage content of 5 metagenomic marine samples. This method allowed us to generate simulated metagenomes that are as close and possible to a real metagenome set while excluding the unknown fraction of the microbial population. The distance between the original taxonomic profile for the sample and the profile used for simulation was calculated as Bray-Curtis dissimilarity (Supplementary Fig B-S4). This computational method allowed us to generate simulated metagenomes containing 5% of phage sequence content and a realistic distribution of contigs length.

The precision and sensitivity of the tools on the *simulated metagenome* were assessed based on the contig length (Fig B-7, and *F*1score in Supplementary Fig B-S4). The observed sensitivity

(Fig B-7A) is consistent with the results from the *simulated phageome set*, although MARVEL displays a large variance in sensitivity across the different replicates, possibly reflecting the quality of binning.

Importantly, using this benchmark set, a decrease in precision due to low viral abundance is clearly seen for all tools (Fig B-7B). VirSorter, viralVerify, VIBRANT, and VirSorter2 perform slightly better than VirFinder, Seeker, and MetaPhinder. ViralVerify performs exceptionally well for long contigs (>= 10^5), followed closely by VirFinder.

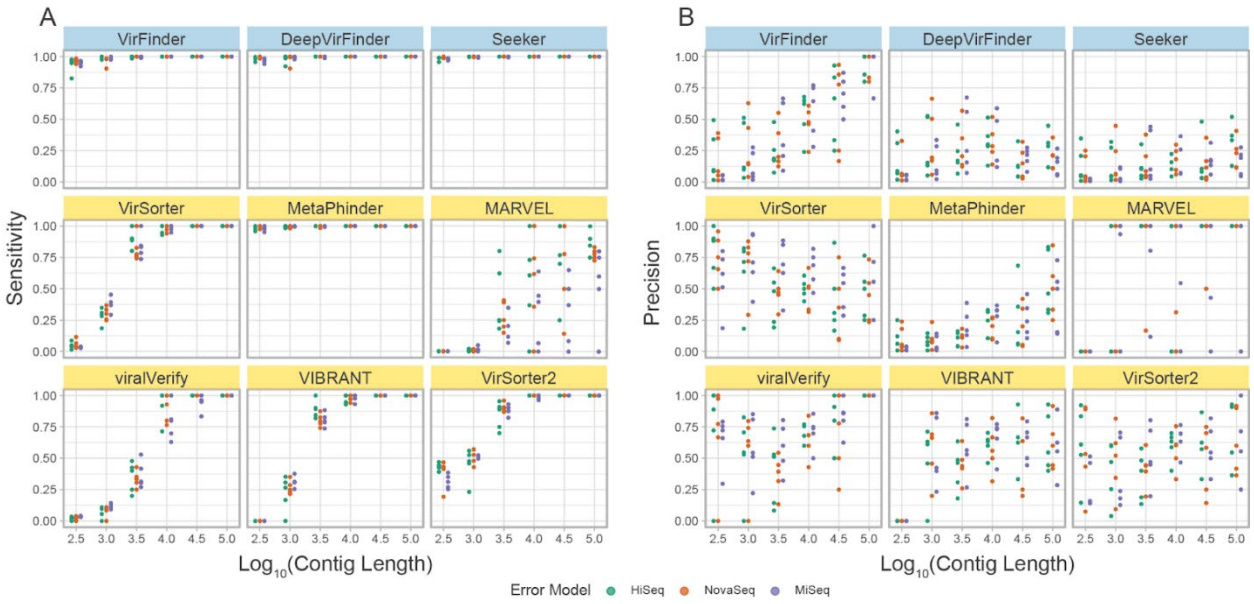

Figure B - 7

Precision and sensitivity obtained while classifying contigs in the *simulated metagenome set*. Each point represents the tool's performance on contigs within a given length group, assembled from reads from a specific abundance profile, using the indicated error model. Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

3.5. Benchmark Challenge 3: Comparison on a Real-World Dataset

We finally compared the tools on a real dataset, the *CRC dataset,* for which the true phage/bacterial composition is unknown. We aimed here to assess the overlap in phage identification of the different tools.

Importantly, the proportions of contigs predicted to be phage vary strongly by tool. VirSorter, MARVEL, viralVerify, VIBRANT, and VirSorter2 detect fewer phages (median less than 2,200 contigs per sample) than the sequence-based methods and MetaPhinder (median greater than 33,000 contigs per sample). This result is consistent with the high precision, but lower sensitivity measured for homology-based tools.

We next assessed the overlap of the different tools in predicting the same sequence as phage (Fig B-8). Strikingly, there was very little overlap in phage communities predicted by the tools (Fig B-8B). The highest level of consistency between tools was seen for VirFinder and DeepVirFinder (38.8% of contigs identified by either tool were identified by both), and the highest level of consistency between homology-based tools was with VirSorter and VIBRANT (26.6% of predicted phages were in common). However, most contigs showed different levels of consistency between tools, where on average, 55,320 contigs were predicted to be phage by only one tool, 62,400 were predicted by 2 or more tools, and 29,900 by 3 or more tools.

CheckV (Nayfach et al. 2021) was used to evaluate the predicted phage contigs for viral genes. Potential viral contigs are categorized as “Not Determined”, “Low Quality”, “Medium Quality”, “High Quality” or “Complete”. The proportions of contigs predicted by the tools falling into each category are shown in Fig B-9C. As expected, the sequence-based tools and MetaPhinder contain a low proportion of contigs with Medium Quality or higher, while the other homology-based tools have higher quality predictions. MARVEL predicted no contigs below Medium

Quality. Additionally, the number of predicted phages being labeled as Medium Quality, High Quality, or Complete by CheckV are shown in Fig B-9D.

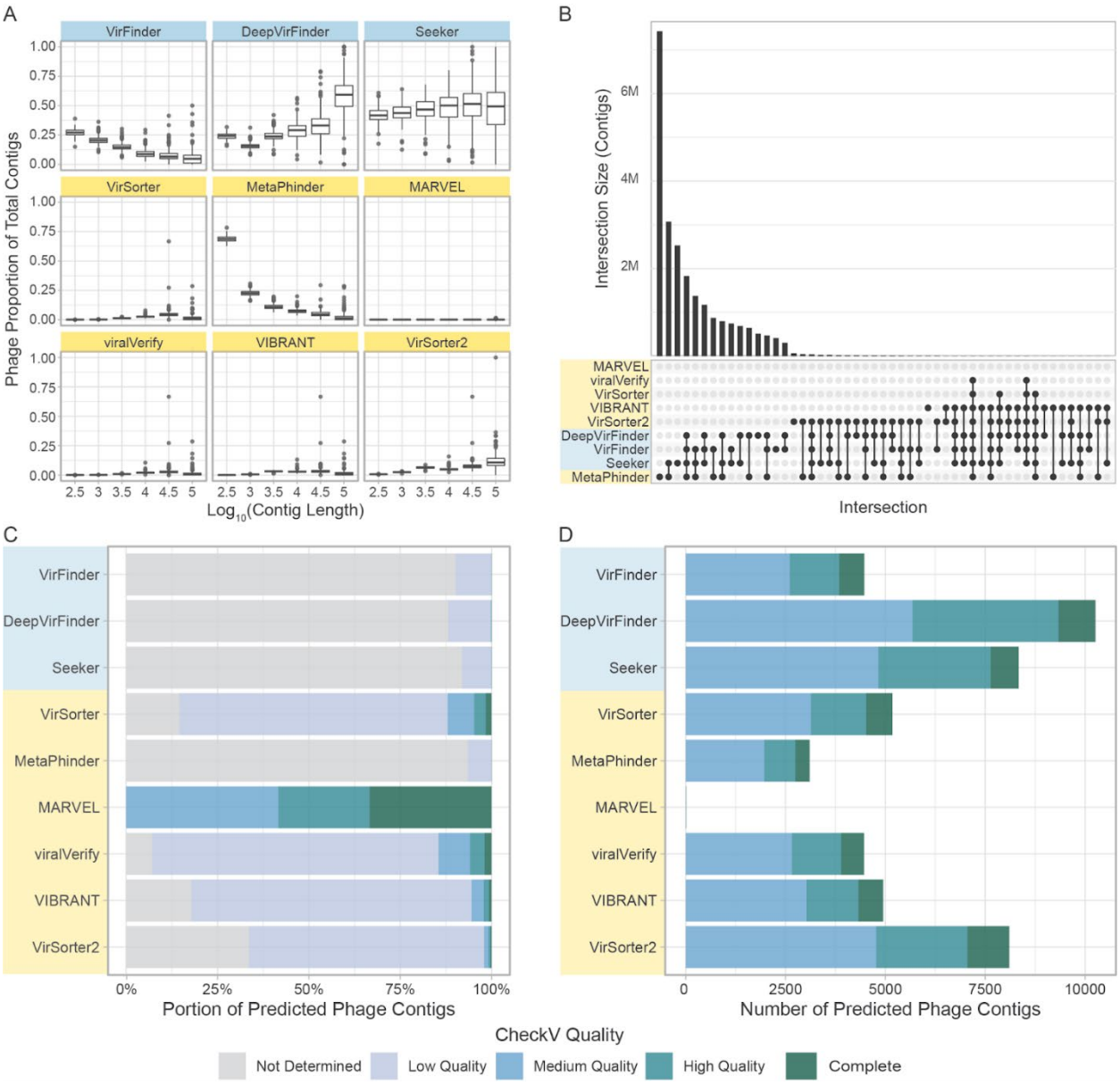

Figure B - 8

Results from classifying *CRC dataset*. A) proportion of contigs predicted to be phage in each length group for each sample. B) upset plot showing intersection size, with the x-axis in order of

decreasing set size, top 51 intersections shown. C) CheckV assessment of predicted phage contigs from the *CRC dataset*. Predicted phage contigs from each tool are categorized by CheckV and plotted as a stacked bar chart of the portion of predicted phages in each category. D) Total number of contigs predicted by CheckV to have quality Medium or greater. Sequence-based tools are in blue, homology-based tools are in yellow.

3.6. Reusable Benchmark Dataset

The three benchmark datasets are available on Zenodo (doi.org/10.5281/zenodo.7194617). Files for the *genome fragment set* include a FASTA file of genome fragments, a CSV file of taxonomic information of each fragment, and a compiled and cleaned file of the classification results of the nine tools on the genome fragments. Files for the *simulated metagenome set* and *simulated phageome set* include the assembly files in FASTA format, the binned assemblies, a CSV file of the contig taxonomic origins resulting from BLAST, and a CSV file of the compiled contig classifications by the tools. For the *CRC dataset*, the assemblies and bins, as well as the compiled contig classifications, are available. For all datasets, the resource usage as recorded by Snakemake is also included.

In addition to README files for each dataset, datasheets based on Datasheets for Datasets (Gebru et al. 2021) are deposited in Zenodo. These provide details about how the datasets were generated, their composition, and the intended uses.

4. Discussion

This study aims to gain a better understanding of how metagenomic phage detection tools perform under a variety of conditions. Previous studies have explored the detection and classification of dsDNA viruses and auxiliary metabolic genes, but the performance of these tools under a variety of challenges remain. Given differences in metagenomes based on the taxonomic

composition, sequencing quality, and computational methods for analyzing these metagenomes, we examined the effects of contig length, phage taxonomy, and sequencing and assembly error. We sought to address the robustness of tools to eukaryotic contamination and low viral content. Finally, we wanted to address the different phage communities predicted by the tools when classifying real metagenomic samples.

4.1. Tools Installation, Reusability, and Computational Requirements

One of the first barriers to the effectiveness of a phage detection tool, and bioinformatics tools in general, is the ability to be installed and scaled to real data. Of the 19 tools identified for this study, only 9 (47%) could be installed and run at scale. This is corroborated by related studies, such as Pratama *et. al.* (Pratama et al. 2021), which excluded PHASTER and VirMiner, and Ho *et. al.* (Ho, Millard, and Schaik 2021), which excluded VIROME, VirMiner, ViraMiner, PhaMers, VirNet, and VirMine from their benchmarking efforts. Tools that were available on Bioconda and PyPi were easiest to install because the tool and all dependencies could be installed simultaneously, while those on GitHub required more effort. We acknowledge the extra effort required to develop tools and create releases using Bioconda and PyPi.

There have been efforts to make tools more accessible, even if the original version is not so straightforward to install. One solution is sharing containers, such as on DockerHub, Biocontainers on the AWS ECR public gallery, or CyVerse ("Docker Hub Container Image Library | App Containerization" n.d.; da Veiga Leprevost et al. 2017; "CyVerse The Open Science Workspace for Collaborative Data-Driven Discovery" n.d.). One effort specific to phage-finding tools is What The Phage, which developed a Nextflow-based workflow for running several phage detection tools in a single container (Marquet et al. 2022).

In addition to the ability to install and run a tool, scalability and resource usage are significant factors considering the scale of metagenomic data. While web services provide a convenient interface, they are often not a viable option for classifying many samples. Similarly, tools that do not allow several samples to be processed at the same are impractical in most cases. For the tools that could be installed and run at scale, computational resource usage varied widely. Homology-based tool compute times generally varied with contig length. Seeker had the most consistently high compute times (greater than 30 hours to classify 10k fragments). While DeepVirFinder had the most consistently low compute times, its inability to process several samples in parallel hindered its actual runtimes. However, this could be circumvented by running each process in its own container, each with its own Theano backend.

4.2. Performance on Short Sequences

In this study, we compared the tools on several challenges to identify the current limitations and strengths of tools and computational approaches. First, we evaluated the effect of sequence length in classification performance. Unsurprisingly, sequence-based tools were able to work with shorter contigs than homology-based tools, as they did not require the presence of multiple genes to classify a sequence. Similarly, the precision of homology-based tool was largely independent of fragment length since shorter contigs would not be expected to affect the false positive rate. Finally, the sensitivity of sequence-based tools such as VirFinder, DeepVirFinder and Seeker were globally unaffected by the sequence length and allow the retrieval of sequences as short as 500 bp. However, a decreased precision for shorter contigs could be observed for these tools, suggesting that a higher number of false positives should be expected when using these tools on shorter contigs. On the other hand, homology-based tools such as VirSorter2, Vibrant and viralVerify, showed a reduced sensitivity on short contigs but a consistent precision for all lengths.

Interestingly, Metaphinder, which leverages multiple hits against a genome database, showed a similar effect for contig length as the sequence-based tools, with an increased false positive rate for shorter contigs. Metaphinder sums the regions of the fragment that have BLASTn hits to phage genomes in a reference database, even if the hits are from distinct phage groups. This, coupled with a fairly permissive e-value of 0.05, leads to a sensitive but less precise classification for short fragments.

The result of this reduced sensitivity on short contigs is seen in both the *simulated phageome set* and *simulated metagenome set*, where VirSorter, viralVerify, VIBRANT, and VirSorter2 recover less than 75% of phage contigs between 1,000 bp and ~3,200 bp long. Considering the difficulty of assembling short reads in real metagenomes, this poses a barrier to retrieving a large number of phage sequences present only as short contigs. For perspective, 93% of the contigs in the *CRC dataset* were shorter than 3k bp.

4.3. Bias Toward Over-Represented Phage Groups

The ability of these tools to identify novel phages or phages with lower database representation can be critical for exploring many natural viral populations. We assessed this by comparing the sensitivity of the tools on the well-represented Caudoviral group and other phage groups. Importantly, all tools showed a decreased sensitivity for non-caudoviral phages. This effect was also seen when comparing the sensitivity of tools on the more abundant *Myoviridae* phages compared to the *Siphoviridae* and *Podoviridae* phages. This result is particularly striking as it suggests a bias in sensitivity toward the over-represented phages groups in databases. Of the tools included in this benchmark, DeepVirFinder appears to be the most suited to detecting a wider diversity of phages and showed the most consistent sensitivity across phage groups. It is important

to note here that our benchmark dataset relied on RefSeq genome sequences and is therefore limited to known phages.

4.4. Low Viral Content and Eukaryotic Sequences

Real metagenomes typically contain low levels of viral content and may also carry eukaryotic sequences from the host (*e.g.,* human gut microbiome) or from micro-eukaryotes. While sequences from a eukaryotic host can typically be excluded from the metagenomic dataset before viral detection, this is particularly difficult for micro-eukaryotic sequences. Of the tools compared, none used eukaryotic genomes in their training set, leading to concerns about specificity when faced with eukaryotic contamination. In this benchmark, we showed that sequence-based tools and MetaPhinder exhibit low specificity on fungal sequence fragments, while other homology-based tools remain unscathed.

To understand how low viral content affects precision, the precision of tools was extrapolated to the full range of possible phage content. All tools had decreased precision when viral content is low, dropping sharply when viral content is below 20%. viralVerify was the most robust to low viral content, especially on shorter contigs. The consequences of this were seen in the classification of *simulated metagenomes*, each of which had 5% phage. All tools had varying and often low (below 0.5) precision, although viralVerify and VirFinder had good precision for the longest contigs.

4.5. Sequencing Error and Assembly Quality

Using simulated metagenomes, we next aimed to assess the effect of sequencing error and assembly on each tool's performance. This method allowed us to develop more realistic benchmark sets while retaining the possibility to assess the true composition of the set.

First, we evaluated the tools' sensitivity on a *simulated phageome set* composed of simulated phage contigs only and assessing the potential effect of sequencing error and potential misassembly. We showed the global sensitivity to be very similar to that obtained when classifying unmodified genome fragments. This indicates that sequencing error, sequencing technology, and misassembly do not hinder sensitivity significantly, at least with sufficient sequencing depth (simulated phageomes were simulated with 30x coverage).

The *simulated metagenome set* aimed to give the most realistic estimate of real-world performance. The sensitivity of all tools again closely reflected previous results. In particular, the contig length and the low viral content were driving the observed tools' performances. DeepVirFinder, Seeker, and at shorter contig lengths Metaphinder, had precision typically below 0.5. VirSorter, viralVerify, VIBRANT, and VirSorter2 had slightly higher precision, although with high variance, often falling below 0.5. viralVerify, however, was extremely precise for long contigs. Once again, this result suggested a limited effect of sequencing error on the tools' performances.

4.6. Comparing Overlap in Viral Predictions

The previous benchmark challenges suggested vastly different properties that affect the final result obtained by users when using on their real-world datasets. This was further demonstrated here on the *CRC dataset*. The number and predicted phage composition is strikingly different between tools. As expected from the previous benchmarks, the homology-based tools, excluding MetaPhinder, predict far fewer phages than sequence-based tools. But most strikingly, the overlap of sequences found by several tools is surprisingly low. Of all sequences predicted to be phage by at least one tool, only 53% were predicted by two or more tools, and 25% were found by three or more tools, on average. The dissimilarity of contigs predicted as phage by the tools is

so wide that approximately 80% of contigs are predicted to be phage by at least one tool. Consequently, when applying these tools to real datasets, the choice of tool would strongly affect the predicted phage community.

The use of CheckV can help reach a larger level of agreement between tools, when used to filter out potential false positives (Fig B-8D, Supplementary Fig B-S7). When contigs with Low Quality or Not Determined status are removed, then greater than 50% of contigs found by at least one tool are found by 3 or more tools; 26% of contigs found by at least one tool are found by 6 or more tools. However, CheckV is stringent, and additional contigs can be recovered by supplementing the dataset with contigs that lack genes of cellular origin, especially for metagenomes with highly novel phage.

4.7. Toward a Reusable Benchmark Dataset for Viral Tool Assessment

All in all, we summarize each tool's performance for each challenge as detailed in Table 2. Given these insights, the remaining question is "What is the best solution to phage detection and prediction?". Unfortunately, answering this important question is not straightforward, especially given the tradeoffs between precision and sensitivity. However, some general guidelines can be used to decide which tools to use. For physically purified viromes (viral enriched particle metagenomes), precision is less of a concern, so one can prioritize sensitivity, and may choose DeepVirFinder, which also has the best sensitivity to non–caudoviral phages. For metagenomes where phages are actively infecting their bacterial hosts, the research question at hand should be the main driver in deciding. To identify novel phages, DeepVirFinder or Metaphinder may be a good choice, although the results should be further confirmed to avoid false positives, such as using CheckV (Nayfach et al. 2021). However, if one is wanting to study the dominant phages

present in an environment and maintain high confidence in phage calling, VirSorter2 or viralVerify may be good choices.

To facilitate further benchmarking of newer tools, we have made all benchmark datasets available. This includes all input files, such as the genome fragments and the assembled simulated metagenomes. Files giving the taxonomic origins of all fragments and contigs are also available, to serve as an answer key when benchmarking new tools. The resulting classifications have been cleaned and compiled into a consistent format, such that classification results can be compared at the fragment/contig level without having to reclassify the input data. Additionally, all code used to analyze the data and generate figures is available for reference on GitHub (https://github.com/hurwitzlab/phage_detection_benchmarks), although modifications would have to be made to incorporate new tools, due to differences in output format, *etc.*

Tool	Speed	Sensitivity	Precision	Phage Diversity	Eukaryote Specificity
VirFinder	16.80	0.9540	0.827	0.734	0.540
DeepVirFinder	256.00	0.9720	0.828	0.839	0.571
Seeker	7.27	0.9820	0.695	0.773	0.106
VirSorter	226.00	0.1200	0.994	0.750	1.000
MetaPhinder	36.40	0.9790	0.738	0.753	0.876
MARVEL	1.07	0.0142	0.157	NA	NA
viralVerify	32.40	0.0628	0.971	0.607	0.992
VIBRANT	26.30	0.0824	0.199	0.512	0.999
VirSorter2	458.00	0.3810	0.979	0.641	0.999

Table B - 2

Breakdown of tool performance. Tools are colored such that homology-based methods are yellow and sequence-based tools are blue. Colors are scaled from the minimum to maximum value in each column. Scales are linear, except for speed which is scaled to log 10 of speed. Speed is the average number of contigs from the simulated metagenome set classified per second. Sensitivity and precision are averages from classifying the simulated metagenomes set. Diverse Phages is the average ratio of sensitivity on non-caudoviral phages vs caudoviral phages. Eukaryote specificity is the average specificity when classifying eukaryotic genome fragments.

4.8. Limitation and Future Directions

This study aimed to set the basis for the development of a fair and reusable benchmark for viral detection tools. However, we wanted to highlight some limitations to the current benchmark. First, we use the default parameters, reference databases, and trained models for all tools. While admittedly these may not be optimal, we believe that it is likely how a large proportion of researchers would use the tools. Second, all genomes used in the *genome fragment set* and simulated datasets were retrieved from RefSeq. This has a significant overlap with the tools' reference databases and training sets; therefore, this study likely underrepresents the diminished performance on broader phage diversity.

Despite these limitations, we hope the developed benchmark may be informative to users and would be further developed to include new computational challenges. It should be noted that the results presented here are limited to those tools that could be installed and run by July 2021, and since then many more tools have been published (we are aware of 3CAC (Pu and Shamir 2021), DeepMicrobeFinder (Hou et al. 2021), INHERIT (Bai et al. 2022), PHAMB (Johansen et al. 2022), PhaMer (Shang et al. 2022), VirMine 2.0 (Johnson and Putonti, n.d.), and virSearcher

(Q. Liu et al. 2022)). With the potential that one of these tools performs better than those studied, it would be useful to benchmark them using the data and methods developed here. Additionally, an important question that was not addressed in this study is how the tools handle plasmid sequences, which is a potential challenge, especially for sequence-based methods.

5. Conflict of Interest

BH holds concurrent appointments as an Associate Professor of Biosystems Engineering at the University Arizona and as an Amazon Scholar. This publication describes work performed at the University Arizona and is not associated with Amazon. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

6. Author Contributions

BH and AP conceptualized the project; KS and AP conducted data collection and preparation; KS, AP, and JG planned and carried out computational experiments and analyzed the data; KS, AP, and BH wrote the manuscript; All authors contributed to manuscript revision, read, and approved the submitted version.

7. Funding

This work was supported by grants from Academy of Finland [339172 to AP] and Gordon and Betty Moore Foundation [GBMF 8751 to BH]. KS acknowledges funding from a PhD training grant through the U.S. National Institutes of Health T32 GM 132008. JG acknowledges funding from the MARC undergraduate training grant through the U.S. National Institutes of Health T34 GM 8718.

8. Acknowledgements

We thank Dr. Timothy Secomb for financial and scientific support for this study. We thank Dr. Heidi Imker, Dr. Chuck Cook, and the Global Biodata Coalition for their support of this project. We thank all the members of the Hurwitz Lab for fruitful discussions and their scientific support. We thank Dr. Kattika Kaarj for fruitful discussions and help in reviewing this manuscript.

9. References

Abdelkareem, A. O., M. I. Khalil, M. Elaraby, H. Abbas, and A. H. A. Elbehery. 2018. "VirNet: Deep Attention Model for Viral Reads Identification." In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 623–26. https://doi.org/10.1109/ICCES.2018.8639400.

Amgarten, Deyvid, Lucas P. P. Braga, Aline M. da Silva, and João C. Setubal. 2018. "MARVEL, a Tool for Prediction of Bacteriophage Sequences in Metagenomic Bins." *Frontiers in Genetics* 9. https://doi.org/10.3389/fgene.2018.00304.

Andrews, Simon and others. 2010. "FastQC: A Quality Control Tool for High Throughput Sequence Data." Babraham Bioinformatics, Babraham Institute, Cambridge, United Kingdom.

Antipov, Dmitry, Mikhail Raiko, Alla Lapidus, and Pavel A Pevzner. 2020. "MetaviralSPAdes: Assembly of Viruses from Metagenomic Data." *Bioinformatics* 36 (14): 4126–29. https://doi.org/10.1093/bioinformatics/btaa490.

Auslander, Noam, Ayal B Gussow, Sean Benler, Yuri I Wolf, and Eugene V Koonin. 2020. "Seeker: Alignment-Free Identification of Bacteriophage Genomes by Deep Learning." *Nucleic Acids Research* 48 (21): e121–e121. https://doi.org/10.1093/nar/gkaa856.

"Babraham Bioinformatics - Trim Galore!" n.d. Accessed September 20, 2022. https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/.

Bai, Zeheng, Yao-zhong Zhang, Satoru Miyano, Rui Yamaguchi, Kosuke Fujimoto, Satoshi Uematsu, and Seiya Imoto. 2022. "Identification of Bacteriophage Genome Sequences with Representation Learning." *Bioinformatics*, August, btac509. https://doi.org/10.1093/bioinformatics/btac509.

Blazanin, Michael, and Paul E. Turner. 2021. "Community Context Matters for Bacteria-Phage Ecology and Evolution." *The ISME Journal* 15 (11): 3119–28. https://doi.org/10.1038/s41396-021-01012-x.

Breitbart, Mya, Chelsea Bonnain, Kema Malki, and Natalie A. Sawaya. 2018. "Phage Puppet Masters of the Marine Microbial Realm." *Nature Microbiology* 3 (7): 754–66. https://doi.org/10.1038/s41564-018-0166-y.

Breitbart, Mya, and Forest Rohwer. 2005. "Here a Virus, There a Virus, Everywhere the Same Virus?" *Trends in Microbiology* 13 (6): 278–84. https://doi.org/10.1016/j.tim.2005.04.003.

Breitbart, Mya, Peter Salamon, Bjarne Andresen, Joseph M. Mahaffy, Anca M. Segall, David Mead, Farooq Azam, and Forest Rohwer. 2002. "Genomic Analysis of Uncultured Marine Viral Communities." *Proceedings of the National Academy of Sciences* 99 (22): 14250–55. https://doi.org/10.1073/pnas.202488399.

"CyVerse The Open Science Workspace for Collaborative Data-Driven Discovery." n.d. CyVerse. Accessed September 23, 2022. https://cyverse.org/.

Deaton, Jonathan, Feiqiao Brian Yu, and Stephen R. Quake. 2017. “PhaMers Identifies Novel Bacteriophage Sequences from Thermophilic Hot Springs.” *BioRxiv*, July, 169672. https://doi.org/10.1101/169672.

“Docker Hub Container Image Library | App Containerization.” n.d. Accessed September 23, 2022. https://hub.docker.com/.

Edlund, Anna, Tasha M. Santiago-Rodriguez, Tobias K. Boehm, and David T. Pride. 2015. “Bacteriophage and Their Potential Roles in the Human Oral Cavity.” *Journal of Oral Microbiology* 7 (1): 27423. https://doi.org/10.3402/jom.v7.27423.

Fang, Zhencheng, Jie Tan, Shufang Wu, Mo Li, Congmin Xu, Zhongjie Xie, and Huaiqiu Zhu. 2019. “PPR-Meta: A Tool for Identifying Phages and Plasmids from Metagenomic Fragments Using Deep Learning.” *GigaScience* 8 (giz066). https://doi.org/10.1093/gigascience/giz066.

Fuhrman, Jed A. 1999. “Marine Viruses and Their Biogeochemical and Ecological Effects.” *Nature* 399 (6736): 541–48. https://doi.org/10.1038/21119.

Garretto, Andrea, Thomas Hatzopoulos, and Catherine Putonti. 2019. “VirMine: Automated Detection of Viral Sequences from Complex Metagenomic Samples.” *PeerJ* 7 (April): e6695. https://doi.org/10.7717/peerj.6695.

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. “Datasheets for Datasets.” *ArXiv:1803.09010 [Cs]*, December. http://arxiv.org/abs/1803.09010.

Gourlé, Hadrien, Oskar Karlsson-Lindsjö, Juliette Hayer, and Erik Bongcam-Rudloff. 2019. “Simulating Illumina Metagenomic Data with InSilicoSeq.” *Bioinformatics* 35 (3): 521–22. https://doi.org/10.1093/bioinformatics/bty630.

Guo, Jiarong, Ben Bolduc, Ahmed A. Zayed, Arvind Varsani, Guillermo Dominguez-Huerta, Tom O. Delmont, Akbar Adjie Pratama, et al. 2021. “VirSorter2: A Multi-Classifier, Expert-Guided Approach to Detect Diverse DNA and RNA Viruses.” *Microbiome* 9 (1): 37. https://doi.org/10.1186/s40168-020-00990-y.

Ho, Siu Fung Stanley, Andrew D. Millard, and Willem van Schaik. 2021. “Comprehensive Benchmarking of Tools to Identify Phages in Metagenomic Shotgun Sequencing Data.” https://doi.org/10.1101/2021.04.12.438782.

Ho, Siu Fung Stanley, Nicole Wheeler, Andrew D. Millard, and Willem van Schaik. 2022. “Gauge Your Phage: Benchmarking of Bacteriophage Identification Tools in Metagenomic Sequencing Data.” bioRxiv. https://doi.org/10.1101/2021.04.12.438782.

Hou, Shengwei, Siliangyu Cheng, Ting Chen, Jed A. Fuhrman, and Fengzhu Sun. 2021. “DeepMicrobeFinder Sorts Metagenomes into Prokaryotes, Eukaryotes and Viruses, with Marine Applications.” bioRxiv. https://doi.org/10.1101/2021.10.26.466018.

Hurwitz, Bonnie L, and Jana M U’Ren. 2016. “Viral Metabolic Reprogramming in Marine Ecosystems.” *Current Opinion in Microbiology*, Environmental microbiology * Special Section: Megaviromes, 31 (June): 161–68. https://doi.org/10.1016/j.mib.2016.04.002.

Johansen, Joachim, Damian R. Plichta, Jakob Nybo Nissen, Marie Louise Jespersen, Shiraz A. Shah, Ling Deng, Jakob Stokholm, et al. 2022. “Genome Binning of Viral Entities from Bulk Metagenomics Data.” *Nature Communications* 13 (1): 965. https://doi.org/10.1038/s41467-022-28581-5.

Johnson, Genevieve, and Catherine Putonti. n.d. “VirMine 2.0: Identifying Viral Sequences in Microbial Communities.” *Microbiology Resource Announcements* 0 (0): e00107-22. https://doi.org/10.1128/mra.00107-22.

Jurtz, Vanessa Isabell, Julia Villarroel, Ole Lund, Mette Voldby Larsen, and Morten Nielsen. 2016. “MetaPhinder—Identifying Bacteriophage Sequences in Metagenomic Data Sets.” *PLOS ONE* 11 (9): e0163111. https://doi.org/10.1371/journal.pone.0163111.

Kang, Dongwan D., Feng Li, Edward Kirton, Ashleigh Thomas, Rob Egan, Hong An, and Zhong Wang. 2019. “MetaBAT 2: An Adaptive Binning Algorithm for Robust and Efficient Genome Reconstruction from Metagenome Assemblies.” *PeerJ* 7 (July): e7359. https://doi.org/10.7717/peerj.7359.

Karl, David M., and Roger Lukas. 1996. “The Hawaii Ocean Time-Series (HOT) Program: Background, Rationale and Field Implementation.” *Deep Sea Research Part II: Topical Studies in Oceanography* 43 (2): 129–56. https://doi.org/10.1016/0967-0645(96)00005-7.

Kieft, Kristopher, Zhichao Zhou, and Karthik Anantharaman. 2020. “VIBRANT: Automated Recovery, Annotation and Curation of Microbial Viruses, and Evaluation of Viral Community Function from Genomic Sequences.” *Microbiome* 8 (1): 90. https://doi.org/10.1186/s40168-020-00867-0.

Köster, Johannes, and Sven Rahmann. 2012. “Snakemake—a Scalable Bioinformatics Workflow Engine.” *Bioinformatics* 28 (19): 2520–22. https://doi.org/10.1093/bioinformatics/bts480.

Langmead, Ben, and Steven L. Salzberg. 2012. “Fast Gapped-Read Alignment with Bowtie 2.” *Nature Methods* 9 (4): 357–59. https://doi.org/10.1038/nmeth.1923.

Li, Dinghua, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. 2015. “MEGAHIT: An Ultra-Fast Single-Node Solution for Large and Complex Metagenomics Assembly via Succinct de Bruijn Graph.” *Bioinformatics* 31 (10): 1674–76. https://doi.org/10.1093/bioinformatics/btv033.

Li, Yuhao, Scott A. Handley, and Megan T. Baldridge. 2021. "The Dark Side of the Gut: Virome–Host Interactions in Intestinal Homeostasis and Disease." *Journal of Experimental Medicine* 218 (5): e20201044. https://doi.org/10.1084/jem.20201044.

Liu, F., Y. Miao, Y. Liu, and T. Hou. 2020. "RNN-VirSeeker: A Deep Learning Method for Identification of Short Viral Sequences from Metagenomes." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1. https://doi.org/10.1109/TCBB.2020.3044575.

Liu, Qiaoliang, Fu Liu, Yan Miao, Jiaxue He, Tian Dong, Tao Hou, and Yun Liu. 2022. "VirSearcher: Identifying Bacteriophages from Metagenomes by Combining Convolutional Neural Network and Gene Information." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1. https://doi.org/10.1109/TCBB.2022.3161135.

Lu, Jennifer, Florian P. Breitwieser, Peter Thielen, and Steven L. Salzberg. 2017. "Bracken: Estimating Species Abundance in Metagenomics Data." *PeerJ Computer Science* 3 (January): e104. https://doi.org/10.7717/peerj-cs.104.

Manrique, Pilar, Michael Dills, and Mark J. Young. 2017. "The Human Gut Phage Community and Its Implications for Health and Disease." *Viruses* 9 (6): 141. https://doi.org/10.3390/v9060141.

Marquet, Mike, Martin Hölzer, Mathias W. Pletz, Adrian Viehweger, Oliwia Makarewicz, Ralf Ehricht, and Christian Brandt. 2022. "What the Phage: A Scalable Workflow for the Identification and Analysis of Phage Sequences." bioRxiv. https://doi.org/10.1101/2020.07.24.219899.

McElroy, Kerensa E., Fabio Luciani, and Torsten Thomas. 2012. “GemSIM: General, Error-Model Based Simulator of next-Generation Sequencing Data.” *BMC Genomics* 13 (1): 74. https://doi.org/10.1186/1471-2164-13-74.

Meyer, Fernando, Andreas Bremges, Peter Belmann, Stefan Janssen, Alice C. McHardy, and David Koslicki. 2019. “Assessing Taxonomic Metagenome Profilers with OPAL.” *Genome Biology* 20 (1): 51. https://doi.org/10.1186/s13059-019-1646-y.

Nayfach, Stephen, Antonio Pedro Camargo, Frederik Schulz, Emiley Eloe-Fadrosh, Simon Roux, and Nikos C. Kyrpides. 2021. “CheckV Assesses the Quality and Completeness of Metagenome-Assembled Viral Genomes.” *Nature Biotechnology* 39 (5): 578–85. https://doi.org/10.1038/s41587-020-00774-7.

Ofir, Gal, and Rotem Sorek. 2018. “Contemporary Phage Biology: From Classic Models to New Insights.” *Cell* 172 (6): 1260–70. https://doi.org/10.1016/j.cell.2017.10.045.

O’Leary, Nuala A., Mathew W. Wright, J. Rodney Brister, Stacy Ciufo, Diana Haddad, Rich McVeigh, Bhanu Rajput, et al. 2016. “Reference Sequence (RefSeq) Database at NCBI: Current Status, Taxonomic Expansion, and Functional Annotation.” *Nucleic Acids Research* 44 (D1): D733-745. https://doi.org/10.1093/nar/gkv1189.

Ponsero, Alise J., and Bonnie L. Hurwitz. 2019. “The Promises and Pitfalls of Machine Learning for Detecting Viruses in Aquatic Metagenomes.” *Frontiers in Microbiology* 10. https://www.frontiersin.org/article/10.3389/fmicb.2019.00806.

Pratama, Akbar Adjie, Benjamin Bolduc, Ahmed A. Zayed, Zhi-Ping Zhong, Jiarong Guo, Dean R. Vik, Maria Consuelo Gazitúa, James M. Wainaina, Simon Roux, and Matthew B. Sullivan. 2021. “Expanding Standards in Viromics: In Silico Evaluation of DsDNA Viral

Genome Identification, Classification, and Auxiliary Metabolic Gene Curation." *PeerJ* 9 (June): e11447. https://doi.org/10.7717/peerj.11447.

Pu, Lianrong, and Ron Shamir. 2021. "3CAC: Improving the Classification of Phages and Plasmids from Metagenomic Assemblies Using Assembly Graph." https://doi.org/10.1101/2021.11.05.467408.

Ren, Jie, Nathan A. Ahlgren, Yang Young Lu, Jed A. Fuhrman, and Fengzhu Sun. 2017. "VirFinder: A Novel k-Mer Based Tool for Identifying Viral Sequences from Assembled Metagenomic Data." *Microbiome* 5 (1): 69. https://doi.org/10.1186/s40168-017-0283-5.

Ren, Jie, Kai Song, Chao Deng, Nathan A. Ahlgren, Jed A. Fuhrman, Yi Li, Xiaohui Xie, Ryan Poplin, and Fengzhu Sun. 2020. "Identifying Viruses from Metagenomic Data Using Deep Learning." *Quantitative Biology* 8 (1): 64–77. https://doi.org/10.1007/s40484-019-0187-4.

Richter, Daniel C., Felix Ott, Alexander F. Auch, Ramona Schmid, and Daniel H. Huson. 2008. "MetaSim—A Sequencing Simulator for Genomics and Metagenomics." *PLOS ONE* 3 (10): e3373. https://doi.org/10.1371/journal.pone.0003373.

Roach, Michael J., Katelyn McNair, Maciej Michalczyk, Sarah K. Giles, Laura K. Inglis, Evan Pargin, Jakub Barylski, Simon Roux, Przemysław Decewicz, and Robert A. Edwards. 2022. "Philympics 2021: Prophage Predictions Perplex Programs." F1000Research. https://doi.org/10.12688/f1000research.54449.2.

Roux, Simon, Francois Enault, Bonnie L. Hurwitz, and Matthew B. Sullivan. 2015. "VirSorter: Mining Viral Signal from Microbial Genomic Data." *PeerJ* 3 (May): e985. https://doi.org/10.7717/peerj.985.

Satinsky, Brandon M, Brian L Zielinski, Mary Doherty, Christa B Smith, Shalabh Sharma, John H Paul, Byron C Crump, and Mary Ann Moran. 2014. “The Amazon Continuum Dataset: Quantitative Metagenomic and Metatranscriptomic Inventories of the Amazon River Plume, June 2010.” *Microbiome* 2 (1): 1–7.

Sczyrba, Alexander, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Dröge, Ivan Gregor, et al. 2017. “Critical Assessment of Metagenome Interpretation—a Benchmark of Metagenomics Software.” *Nature Methods* 14 (11): 1063–71. https://doi.org/10.1038/nmeth.4458.

Shang, Jiayu, Xubo Tang, Ruocheng Guo, and Yanni Sun. 2022. “Accurate Identification of Bacteriophages from Metagenomic Data Using Transformer.” *ArXiv:2201.04778 [q-Bio]*, January. http://arxiv.org/abs/2201.04778.

Sharma, Neetu, Sonu Bhatia, Abhinashi Singh Sodhi, and Navneet Batra. 2018. “Oral Microbiome and Health.” *AIMS Microbiology* 4 (1): 42–66. https://doi.org/10.3934/microbiol.2018.1.42.

Tampuu, Ardi, Zurab Bzhalava, Joakim Dillner, and Raul Vicente. 2019. “ViraMiner: Deep Learning on Raw DNA Sequences for Identifying Viral Genomes in Human Samples.” *PLOS ONE* 14 (9): e0222271. https://doi.org/10.1371/journal.pone.0222271.

Tisza, Michael J, Anna K Belford, Guillermo Domínguez-Huerta, Benjamin Bolduc, and Christopher B Buck. 2021. “Cenote-Taker 2 Democratizes Virus Discovery and Sequence Annotation.” *Virus Evolution* 7 (1): veaa100. https://doi.org/10.1093/ve/veaa100.

Turner, Dann, Andrew M. Kropinski, and Evelien M. Adriaenssens. 2021. “A Roadmap for Genome-Based Phage Taxonomy.” *Viruses* 13 (3): 506. https://doi.org/10.3390/v13030506.

Veiga Leprevost, Felipe da, Björn A Grüning, Saulo Alves Aflitos, Hannes L Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, et al. 2017. “BioContainers: An Open-Source and Community-Driven Framework for Software Standardization.” *Bioinformatics* 33 (16): 2580–82. https://doi.org/10.1093/bioinformatics/btx192.

Wommack, K. Eric, Jaysheel Bhavsar, Shawn W. Polson, Jing Chen, Michael Dumas, Sharath Srinivasiah, Megan Furman, Sanchita Jamindar, and Daniel J. Nasko. 2012. “VIROME: A Standard Operating Procedure for Analysis of Viral Metagenome Sequences.” *Standards in Genomic Sciences* 6 (3): 427–39. https://doi.org/10.4056/sigs.2945050.

Wood, Derrick E., Jennifer Lu, and Ben Langmead. 2019. “Improved Metagenomic Analysis with Kraken 2.” *Genome Biology* 20 (1): 257. https://doi.org/10.1186/s13059-019-1891-0.

Yoo, Andy B., Morris A. Jette, and Mark Grondona. 2003. “SLURM: Simple Linux Utility for Resource Management.” In *Job Scheduling Strategies for Parallel Processing*, edited by Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, 44–60. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/10968987_3.

Yu, Zhenhua, Fang Du, Rongjun Ban, and Yuanwei Zhang. 2020. “SimuSCoP: Reliably Simulate Illumina Sequencing Data Based on Position and Context Dependent Profiles.” *BMC Bioinformatics* 21 (1): 331. https://doi.org/10.1186/s12859-020-03665-5.

Zhao, Guoyan, Guang Wu, Efrem S. Lim, Lindsay Droit, Siddharth Krishnamurthy, Dan H. Barouch, Herbert W. Virgin, and David Wang. 2017. “VirusSeeker, a Computational

Pipeline for Virus Discovery and Virome Composition Analysis." *Virology* 503 (March): 21–30. https://doi.org/10.1016/j.virol.2017.01.005.

Zhao, Min, Di Liu, and Hong Qu. 2017. "Systematic Review of Next-Generation Sequencing Simulators: Computational Tools, Features and Perspectives." *Briefings in Functional Genomics* 16 (3): 121–28. https://doi.org/10.1093/bfgp/elw012.

Zheng, Tingting, Jun Li, Yueqiong Ni, Kang Kang, Maria-Anna Misiakou, Lejla Imamovic, Billy K. C. Chow, et al. 2019. "Mining, Analyzing, and Integrating Viral Signals from Metagenomic Data." *Microbiome* 7 (1): 42. https://doi.org/10.1186/s40168-019-0657-y.

10. Supplementary Materials

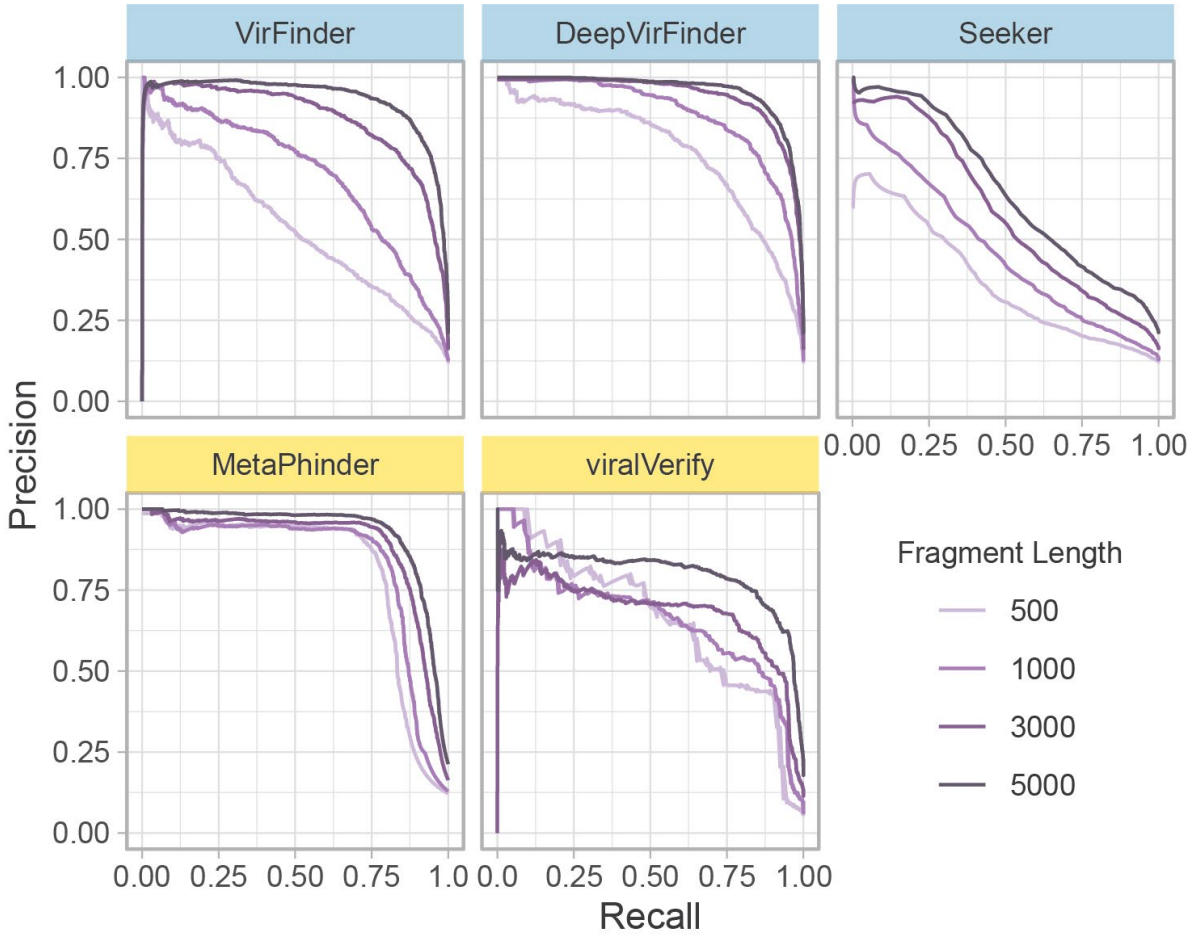

Figure B S - 1

Precision-Recall curves of tools for which a predicted value is output. Only bacterial and phage fragments are included. Top row of tools are sequence-based (in blue), bottom two row is homology-based (in yellow).

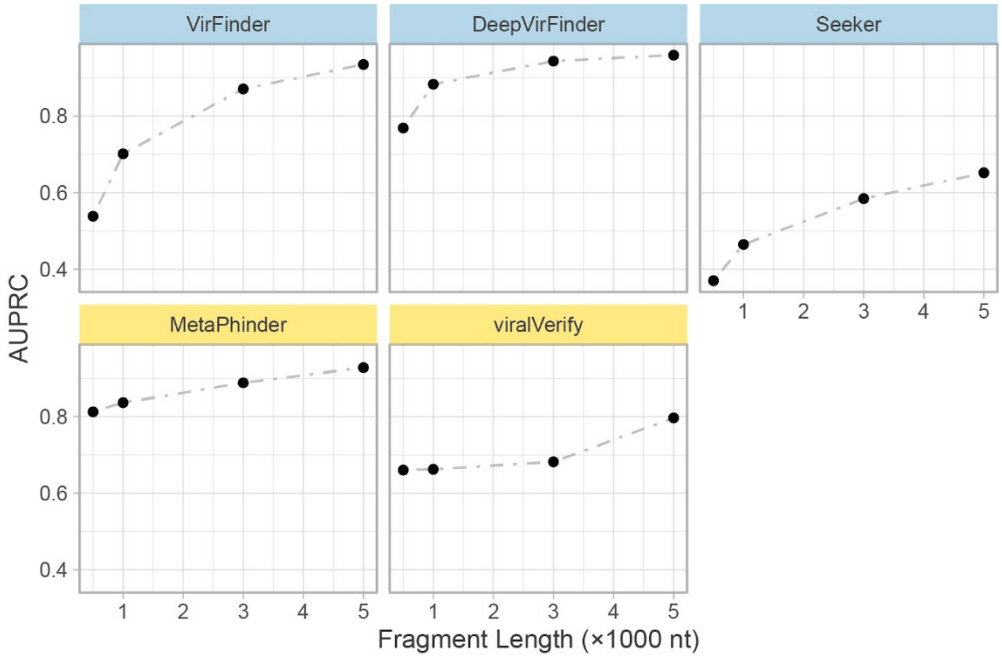

Figure B S - 2

Area under precision recall curve (AUPRC) vs fragment length of tools for which a predicted value is output. Only bacterial and phage fragments are included. Top row of tools are sequence-based (in blue), bottom row are homology-based (in yellow).

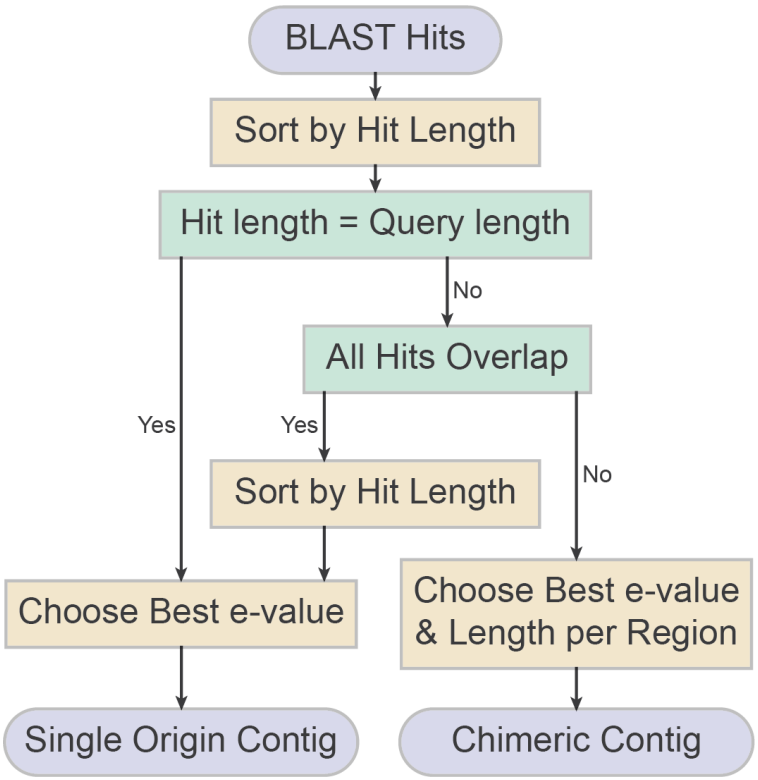

Figure B S - 3

Flowchart decision tree of how taxonomy of simulated contigs were assigned based on BLASTn search against a database of genomes used for read simulation.

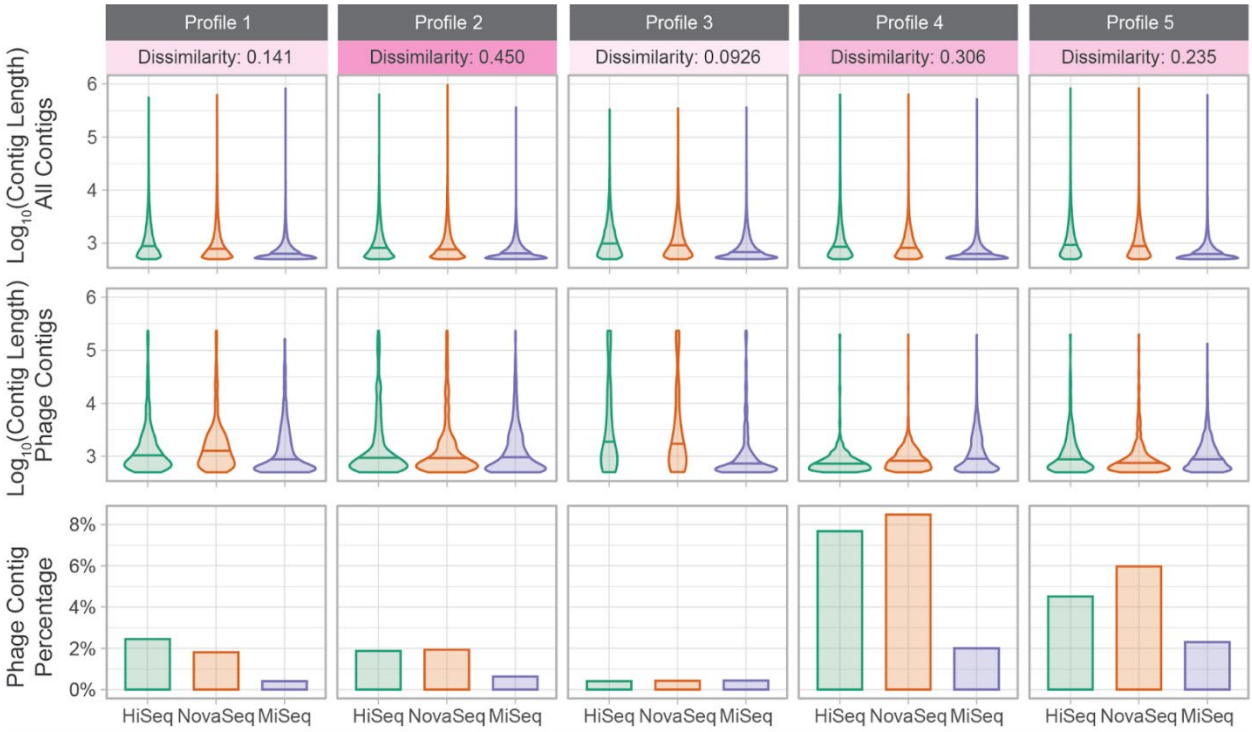

Figure B S - 4

Characteristics of simulated metagenomes generated by InSilicoSeq based on 5 marine abundance profiles using the 3 built-in error models (HiSeq, MiSeq, and NovaSeq). From top to bottom: Bray-Curtis dissimilarity between the original marine abundance profile and the abundance profile used for read simulation; Violin plots of (top row) all contigs, and (middle row) phage contig length distributions on log 10 scale with constant area violins and median marked with a horizontal line; Percentage of contigs that are from phages (bottom row).

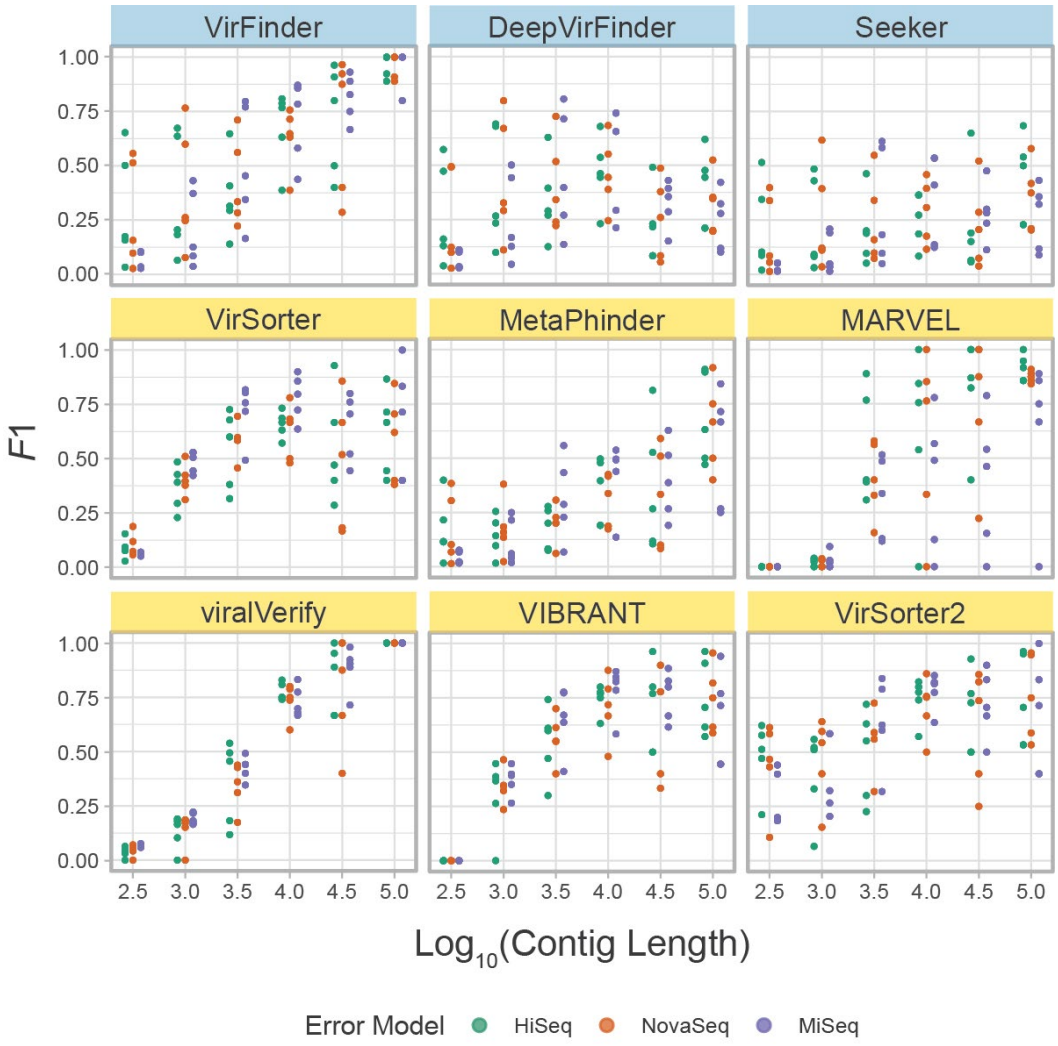

Figure B S - 5

*F*1 score for the classification of the simulated metagenome set vs log 10 of contig length. Top row of tools are sequence-based (in blue), bottom two rows are homology-based (in yellow).

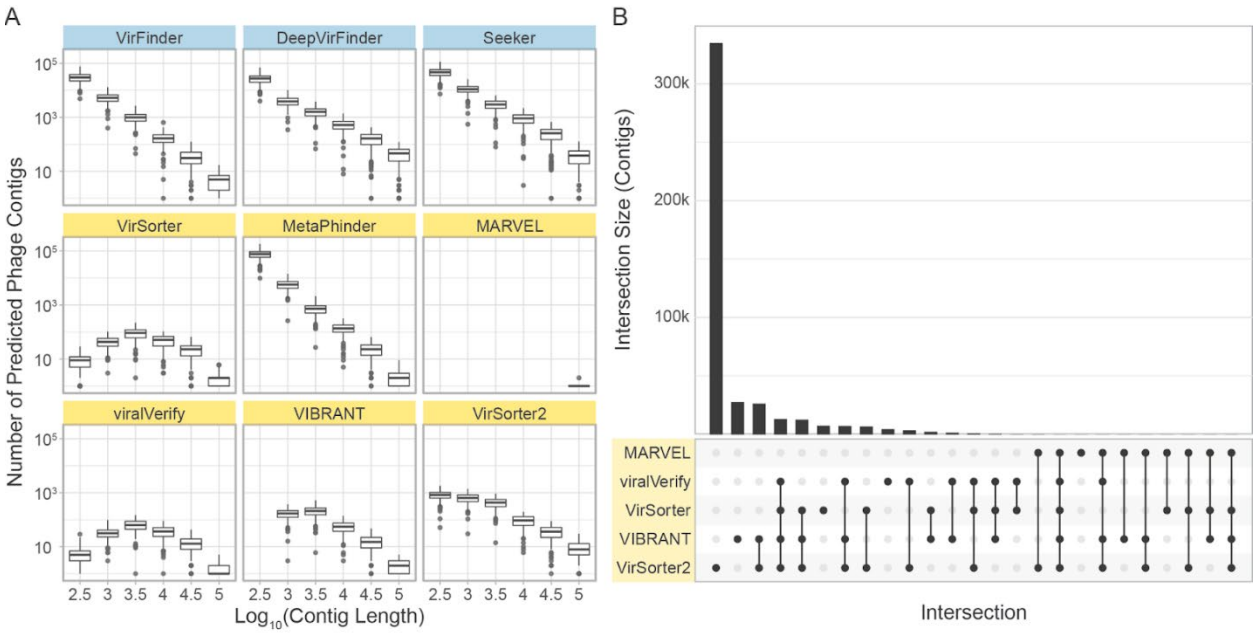

Figure B S - 6

Results of classifying *CRC dataset* A) Boxplot of the number of predicted phage contigs, on a log scale, in each length group per sample. B) Upset plot of homology-based tools, excluding MetaPhinder, showing intersection sizes in decreasing order. Homology-based tools are in yellow, sequence-based tools are in blue.

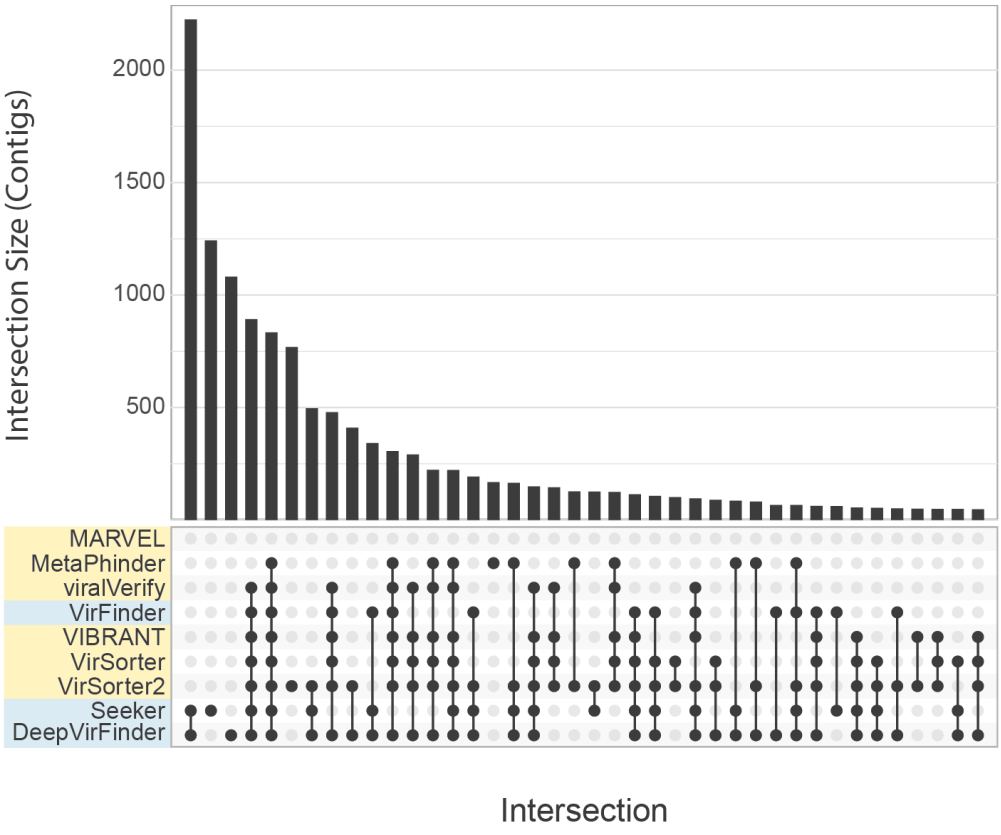

Figure B S - 7

Results from filtering predicted phage contigs, by including only those contigs labeled by CheckV as having Medium Quality, High Quality, or Complete virus. Upset plot showing the intersection sizes between the tools for contigs in one of those categories. Homology-based tools are in yellow, sequence-based tools are in blue.

APPENDIX C

A Detailed Case Study on Implementing a Reproducible Machine Learning Enabled Workflow

Kenneth E. Schackart III[1,2,*], Heidi J. Imker[2,3], Chuck Cook[2]

[1]Department of Biosystems Engineering, The University of Arizona, Tucson, Arizona 85721, USA

[2]Global Biodata Coalition, Strasbourg, France

[3]University Library, University of Illinois at Urbana-Champaign, Urbana, Illinois 61821, USA

[*]Correspondence: schackartk1@gmail.com

Manuscript in Preparation

Abstract

Machine learning (ML) and advanced computational methods are being developed and applied at a rapid pace. While these methods provide powerful ways of processing large amounts of data to gain valuable insights, best practices are still evolving. Uncodified code and data standards have led to the irreproducibility of ML-enabled research. Improved standards have been proposed, but researchers who wish to adopt these standards may be unsure how to meet them. In this Comment, we describe an ML-enabled research project to create a global inventory of biodata resources, using it as a case study to discuss the efforts and decisions we made to make this study reproducible. We discuss the software libraries, tools, and practices we used to obtain high standards for reproducibility, code, and data. This work serves as an example for others and encourages discussion on how the approach can be improved.

1. Introduction

Science hinges on the ability to test hypotheses. This means that the hypothesis must be falsifiable and supporting results can be reproduced. When research findings cannot be reproduced, they must be called into question. Since Ioannidis published the paper "Why Most Published Research Findings Are False", there has been broad concern over the lack of reproducibility in science (Ioannidis, 2005). This phenomenon has been termed the "reproducibility crisis", although this has been contested (Fanelli, 2018). Concerns about scientific reproducibility are ongoing, and flawed study designs and irreproducible analyses play a role. There have been efforts to encourage better practices, such as pre-publication of study protocols, analysis plans, and code (Haring & Bell, 2018). However, as argued in Haring 2018, while the different biases in production and reporting of research are largely identifiable and modifiable, continued methodological training for early career researchers is also crucial.

Use of machine learning (ML) in biosciences has outpaced the adoption of good practices and proper training. Open Science practices, such as public release of code and data, aim to remedy this. While access to code and data are necessary for reproduction of computational results, it does not guarantee the results can be reproduced. Efforts have been made to set standards for reproducible code, including for ML, and they serve as great rubrics for assessing reproducibility (Heil et al., 2021). What is lacking, however, are detailed examples of practical implementation. This work aims to provide such an example by explaining how an ML enabled study was planned and executed with reproducibility as an explicit goal from the onset of the project

The study under question is a ML enabled inventory of global biodata resources from the scientific literature REF. The study was undertaken because these resources are distributed all across the world but are difficult to locate in any systematic way, which makes it difficult to describe (let alone understand) the entire infrastructure. Briefly, a targeted query from EuropePMC (The Europe PMC Consortium, 2015) was submitted to retrieve titles and abstracts that contain URLs and word data, with the possibility that they describe a biological data (biodata) resource. A random subset of these were manually classified as describing or not describing a biodata resource. Those that did describe a biodata resource were curated to label the resource's common name (*e.g.,* PDB) and full name (*e.g.,* Protein Data Bank) (Berman et al., 2003). Pretrained transformer models were fine-tuned for the classification (predicting if the article describes a biodata resource) and named-entity recognition (predicting common and full name) tasks. Several pretrained models were fine-tuned and compared. Further downstream processing was performed such as URL extraction and http status checking. A full article describing the study is in preparation, and during its preparation we quickly realized that there are many additional details to share about how we've attempted to address reproducibility–details we wish we had found in

the literature ourselves. Instead of including extraordinarily lengthy supplemental methods to the forthcoming full article, we offer this Comment as a companion.

During the study, a strong emphasis has been placed on Open Science, reproducibility, and robustness of the codebase and documentation for both philosophical reasons (in support of Open Science) and practical reasons (enabling future updating of the inventory). The entire process, from data splitting, model training and selection, to all downstream processing, is encapsulated in a Snakemake workflow (Köster et al., 2012). This allows for the reproduction of the entire analysis with a single command. Strong standards of code quality were developed and are enforced through the use of automated testing. Additionally, significant efforts were made to make all data products Findable, Accessible, Interoperable, and Reusable (FAIR) (Wilkinson et al., 2016)

As we began the project, we immediately turned to the literature for robust examples that we could follow but were unable to locate implementation details that mapped to our project and goals. With the absence of practical examples, we forged ahead. While we recognize that there may be better ways of doing these tasks, we offer this comment as a starting point and example of how to make a computationally heavy study reproducible and open. We provide the reasoning behind the various considerations, which can be applied generally to research projects. We also provide specific examples of how those were realized in this study.

2. Have A Plan

"A goal without a plan is just a wish," wrote Antoine de Saint-Exupéry in *The Little Prince* (Saint-Exupéry, A., 1943). As with any other part of a research project, planning ahead makes the path to achieving reproducibility as smooth as possible. To this end, early in the project we developed an Open Science Implementation Plan. Here, we outlined the goals for reproducibility,

and how we planned to achieve them. These goals were organized into three groups: reproducibility of methods, code standards, and data standards.

By considering these goals early in the project, we explicitly defined what expectations we had for our Open Science goals. Keeping these goals in mind helped ensure that the effort required to obtain these goals was anticipated and considered a core piece of the project. This minimized accumulation of technical debt that would have been time consuming and difficult to address near the end of the project.

3. Reproducibility of Methods

We found the reproducibility standards defined by Heil et. al. useful for ranking reproducibility levels (Heil et al., 2021) In our case, bronze alone was not acceptable (data published and downloadable, models published and downloadable, source code published and downloadable). Obtaining silver was acceptable (bronze + dependencies set up in a single command, key analysis details recorded, analysis components set to deterministic), but the gold standard was our goal (silver + entire analysis reproducible with a single command).

3.1. Meeting the Bronze Standard

The bronze standard of reproducibility is characterized by having the following published and downloadable: all data necessary for reproduction, trained models, and source code (Heil et al., 2021).

Data availability and, more broadly, FAIRness (findability, accessibility, interoperability, and reusability) will be further discussed in a later section. To address the minimum set out in the bronze standard, all data are available for download from the project's Zenodo repository.

Model availability is addressed in a few ways. All the models used in this project were pretrained by other groups and made available on HuggingFaceHub (HFHub). As part of model

training, these pretrained models were fine-tuned to various tasks (sequence classification and token classification). We will make these fine-tuned models available in a Zenodo archive, as well as on HFHub (Wolf et al., 2020, https://huggingface.co/models).

All source code will be stored in two places. First, GitHub serves as a living repository (github, 2020, https://github.com/globalbiodata/inventory_2022). An important aspect of Open Science is providing a place for open discussion (and criticism) of methods. The GitHub Issues system allows for free and open commentary of the computational methods. However, GitHub repositories are not immutable. It is important to have the methods, as described in the original publication, preserved and available. Source code used to obtain the results in the associated publication that describes the entire project (in preparation) will be deposited into the Zenodo archive unmodified.

3.2. Meeting the Silver Standard

The silver standard requires, in addition to those aspects listed in the bronze standard, all dependencies can be installed and set up with a single command, key analysis details are recorded, and all analysis components are deterministic (not random) (Heil et al., 2021).

A common challenge for reproducibility is having simple installation procedures. To reach the silver standard in this regard, we wanted it to be possible to install all dependencies with a single command. For Python-based projects that is often possible with the command “pip install -r requirements.txt”. However, sometimes other dependencies not covered by pip need to be installed. To simplify this step, we utilized Make (GNU Make 42.1). While Make is a powerful tool intended for the control of executable files, we only use it for effectively creating aliases for shell commands. In the case of installation, we provide a Make target called “setup”. By doing so,

the user can simply type “make setup” and shell commands are executed to install all dependencies, including running pip (v21.1.2).

In addition to providing a simple pip install procedure, we created a conda installation procedure. While pip installing on the user level is sufficient in isolated environments, such as Google Colab, it can lead to conflicts on other systems if a virtual environment is not used. Conda (v4.9.1) (anaconda, 2020) provides an isolated environment in which the project specific dependencies are installed. By providing a conda environment description (yaml) file, it is possible to recreate the conda environment in a single command.

The gold standard for dependency management takes it a step further than virtual environments in the form of containers (Heil et al., 2021). With the use of containers, it is ensured that the entire environment is identical for future users. However, containers can sometimes be problematic to use in certain environments and can require fairly technical set-up. We wanted this project to be reusable to the people with a wide range of technical skills and who may not have ready access to robust computational infrastructure. This is especially important when thinking of potential users, who may not be at well-resourced academic institutions, all over the world. Instead, while we are still considering Dockerization as another option (Merkel, 2014) , we designed this project to be run on Google Colab (Bisong, 2019)for its low barrier to entry and its provision of GPUs for free use. Colab does not support common container services such as Docker. However, by providing several options for dependency installation, we hope that future users can find one to suit their needs.

Sufficient documentation of “key analysis details” is subjective. To satisfy this requirement, in addition to an overview README that describes the entire repository, we provide README files in every directory within the repository, which explain what the various

files/scripts are, and how they relate to each other. Since 2021 GitHub supports the use of Mermaid, JavaScript-based diagramming and charting tool, in markdown files, which we leverage to create informative flowcharts illustrating workflow logic (Sveidqvist, 2022)

An often-overlooked key to reproducibility in computational methods, particularly ML methods, is seeding pseudo-random processes such that they are deterministic (Heil et al., 2021; Ahmed & Lofstead, 2022). The random numbers generated by pseudo-random number generators can have significant effects on the trained model and model performance (Ahmed & Lofstead, 2022). So, to make the process reproducible, we added options to use seeding to make the processes deterministic.

3.3. Meeting the Gold Standard

According to Heil, the gold standard implies that the entire analysis can be run with a single command. This goal was addressed with Snakemake (v7.1.1) (Köster & Rahmann, 2012). There are several options for workflow managers, a popular alternative being Nextflow (Di Tomasso et al., 2017) Snakemake facilitates automation through the definition of "rules" or steps that take inputs and generate outputs. By stating what outputs are desired, Snakemake creates a directed acyclic graph of which rules must be executed to create the specified output. For instance, in this project, we specify that we would like the final output file containing the classified articles along with extracted metadata. If the final output is not present, Snakemake executes all necessary steps in the pipeline including data splitting, model training and comparison, classification and NER, and all downstream processing. With the help of a Make alias, the Snakemake workflow for reproducing all results can be run with the single command "make train_and_predict".

It is important to be able to reproduce all results from the raw data to final results, including model training. However, model training is resource intensive, and often requires the use of

specialized hardware such as a graphic processing unit (GPU). If all models must be trained to reproduce results, it may be a barrier to reproducibility. To minimize the computational resources necessary to reproduction, all trained models are also available in the Zenodo repository.

4. Beyond Reproducibility

This research project is unique because it was known from the beginning that someone will need to rerun the analysis periodically. However, strictly speaking, this went beyond reproduction since the underlying data is expected to shift (processing articles that have been published in the meantime), so the methods developed needed to be generalizable. Generalizability benefits from the same considerations as reproducibility but tends to include additional challenges.

We approached generalizability with the same standards as reproducibility and wanted to make updating the inventory possible with a single command. However, we found that it was far more efficient to directly utilize the trained models in the "updating" workflow. In this way, the steps needed for updating were isolated from the larger reproduction workflow to make this process faster and simpler.

5. Code Standards

We've taken the philosophy that the results of a computational research project are no more trustworthy than the code used to produce them.

5.1. Code Formatting

To accomplish true Open Science, accessibility of code should not be limited to code being publicly available. Code should also be readable and well-documentedd for it to be truly accessible. A good first step is to utilize a code formatter, which all modern programming languages have. We used yapf v0.31.0 to format all the Python code in this project (google, 2022). A common alternative is Black (Langa, 2018), which is significantly stricter, leaving fewer

formatting decisions up to the developer. Similarly, Snakemake files were formatted with snakefmt v0.6.0 (Hall & Letcher, 2020), and R files were formatted with styler v1.7.0 (Müller & Walthert, 2022).

5.2. Static Code Checking

Another measure taken to increase code robustness was static code checking. Again, the code checking tools available will depend on the language. We utilize the linters pylint v2.8.2 and flake8 v.3.9.2 (pylint, 2016; Cordasco & Ziade, 2010) to check all Python code to ensure that community code standards are upheld, and to detect code smell (potentially problematic coding patterns). Many of the items that these linters consider can greatly improve code quality and readability. Some examples of considerations of the linters are: line lengths must be limited to predefined thresholds; within any context (e.g. a function) there should not be too many variables, and all functions should have docstrings. These rules, and many others, encourage developers to write cleaner, more readable code.

Additionally, while type annotations are not required in the Python community, we implemented them for many reasons. Type annotations provide built-in documentation by defining the data types of all inputs and outputs of functions. A lesser-discussed benefit of type annotations is that it provides an enhanced integrated development environment (IDE) experience since the IDE has more knowledge of the variables, and can give better help messages, syntax highlighting, and autocompletion. The final benefit of type annotations we considered is the prevention of unforeseen bugs when they are used in conjunction with a static type checker. We used mypy v0.812 (Lehtosalo et al., 2021) to check type compatibility within all our Python code. This can significantly reduce the chances of encountering bugs that occur not at compile time (since Python

is interpreted and dynamically typed), but instead at runtime, which can be more difficult to resolve and may not show up until running the code.

While static code checking has many benefits, programmers need not strictly adhere to all suggestions made by the code checkers. Luckily, most tools are configurable. Importantly, the user can disable certain warnings. For Pylint, this is done with a resource configuration (rc) file. When an rc file is used to modify code checkers, it is important to include them in the code repository, so that when someone else runs the code checks on your code, they see the same results.

5.3. Testing

A crucial software engineering practice that is largely absent from research code is testing. Testing in all its forms, whether it is unit, integration, or end-to-end, defines the specifications of a piece of software and ensures that the software meets those specifications when the tests pass. This has numerous benefits that cannot be understated.

One of the primary benefits is that tests serve is as a contract, which is a form of documentation. A unit test of a function explicitly states what kinds of input are expected and what kinds of outputs will be produced. For documentation, the only thing better than telling what a function does (through comments and docstrings) is showing through tests. While the descriptions provided in docstrings and comments are what the developer intends the software to do, a passing test demonstrates that it indeed does it. Conversely, anything not covered in the test cases is where the contract ends. Tests ensure that the code can do what it says it can do.

From an Open Science perspective, testing is particularly valuable. Not only does it provide more detailed documentation than could ever be provided in an article's methods section, but it facilitates community feedback and contributions. Making changes to software always poses the risk of disrupting previous functionality. When considering applying community feedback or

contributions, this is problematic. However, with strong test coverage, developers can have more confidence that updates are not introducing breaking changes, as long as all previously passed tests still pass. Indeed, they provide a clear avenue of addressing bugs which the community may catch. Developers can add another test case that exposes the bug, then modify the code such that the new test and all previous tests pass. This is effectively amending the contract the tests provide to be more comprehensive. Without tests in place, developers would have to check that the code still behaves as described manually, which is error-prone and many researchers may be hesitant to implement changes suggested by others.

Of course, adding strong test coverage does require more work than, for instance, implementing static code checks or formatting. Without tests, code must be manually assessed to ensure that a given piece of software is able to perform its intended task, and there is a barrier to implementing community feedback. Further, a lack of tests is a form of technical debt, and the price is paid when trying to refactor or fix bugs.

Pytest v6.2.4 was used as a testing framework for all Python code in this project (Krekel et al., 2004). Pytest plugins for flake8, pylint, and mypy are used to include static code checks of each file as part of the test suite (pytest-flake8 v1.0.7, pytest-pylint v0.18.0, pytest-mypy v0.8.1). This makes it so the test suite cannot pass without all static checks passing. Additionally, most functions have associated tests, and most scripts also have end-to-end tests that ensure that they properly reject bad inputs and produce correct output when given good input. While we aim to have good test coverage, some functions and scripts are not comprehensively tested. This is generally the case for functions/scripts that take a very long time to run, such as the actual process of model training. Additionally, the Snakemake workflows developed are not formally tested using

an automated testing framework, although it would be best to do so, and we may implement this at a later time.

5.4. Configurability

Our aim was that the users of code, whether for reproducibility, generalization, or separate implementation, would not need to edit source code to change its behavior within the intended use-cases. Parameters that may change could be supplied as inputs/arguments instead. Often, this means that paths to input files should not be hard-coded but rather passed in when calling a script. In terms of ML projects, this also often applies to hyperparameters.

One solution is to use parameterization extensively, and to make the analyses reproducible by storing the parameters used in configuration (config) files. By doing so, others can see what parameters were used to generate the results. Additionally, it gives future users a clear indication of what parameters are likely okay to change, all without them having to edit any source code.

We store a large number of parameters in config files such as input/output directories, training parameters, and locations of pretrained models. To train a new model and compare their performance to existing models, a new row needs simply be added to a tab-separated config file. The README file in the config/ directory describes the acceptable ranges of values allowed in the config files, such as a description of what kind of models are compatible with the existing workflow.

Snakemake also makes extensive use of config files, and the config files described here are formatted such that Snakemake can utilize them when executing the workflow. So, to change the behavior of the workflow (again, within the expected range of uses), only config files need to be edited.

6. Data Standards

6.1. Source Selection

Both code and data are integral components of this project, and both require consideration for reproducible outcomes. To create an open inventory as a product, we aimed to reuse and create data that aligned with the FAIR (Findable, Accessible, Interoperable, and Reusable) Guiding Principles (Wilkinson et al., 2016). The primary data source needed was bibliographic metadata. There are several commercial sources of bibliographic metadata such as Dimensions (Digital Science), Scopus (Elsevier), and Web of Science (Clarivate Analytics). However, these resources require a subscription which would limit others' ability to reproduce and reuse this work and neither are they openly licensed. Therefore, we opted to use the open metadata available from Europe PMC as the data source for creating the inventory. Although not as exhaustive as the commercial options mentioned, Europe PMC covers a large swath of the life sciences; as of October 2022, high quality, interoperable metadata is available for over 40 million articles. Additionally, Europe PMC offers robust and well-documented APIs that facilitate access and are especially useful for a reproducible pipeline. Although we know that some biodata resources will be missed due to articles being published outside of the ~4000 journals available in Europe PMC, we felt that this tradeoff was justified in order to optimize openness and reproducibility.

6.2. Data Findability and Accessibility

Depending on context, anyone interested in reusing the data from this project may wish to jump in at different points, therefore we offer multiple options. The exact query string we used can be rerun to obtain results from Europe PMC. Additionally, since bibliographic databases may change slightly over time (*e.g.,* records added, removed, or corrected), query results themselves (PMID, title, abstract) may be of use to reproduce our results using the exact same data. There is

also the labeled training data that was used to train the various models and, finally, the primary data product for this project is the GBC Inventory itself. The query string, query results, training data, and the GBC Inventory are all available within the project's GitHub repository and will be archived for long-term preservation and persistent reference in an associated Zenodo deposit when the preprint of the forthcoming full article is posted. Zenodo provides a DOI and relies on the DataCite metadata schema, which allows for the dataset to be found within Zenodo's search interface, DataCite's central metadata store, and via internet search engines such as Google.

6.3. Data Interoperability

For the final GBC Inventory, we retained unique article identifiers (PMIDs), which allows for easy extraction of additional metadata or for access to the full text, when available, from either Europe PMC or PubMed Central. Additionally, we logged URL status codes per specification RFC 9110, extracted countries from author affiliations following ISO 3166, and retained geo coordinates for IP address lookups, when available. While it would have been ideal to include a persistent identifier for the biodata resources located (*e.g.,* ROR ID or DOI), not all resources have such an identifier, which perfectly illustrates the challenge of trying to locate these resources in the first place.

6.4. Data Reusability

In addition to the efforts towards interoperability described above, we also maintained a structured format throughout and have shared the data in CSV files for preservability and ability to be opened and reused easily. These files are accompanied by a plaintext README file which includes a codebook describing each data variable and data collection details and licensing. By using open data from Europe PMC, we can release the data with CC0 licensing, allowing for the

broadest reuse possible. Together, this documentation, the repository's GitHub history, and Zenodo's commitment to long-term archiving all provide provenance.

Finally, to further extend the potential for reuse, we plan to provide identified biodata resources to Europe PMC as community annotations. This will allow for easy bulk access to the identified resources and their associated articles. The annotations can be used for several purposes, for example, mining articles with full text available or analysis of the intersection between these annotations and the many other annotation types available within Europe PMC.

7. Discussion

Here we have described the efforts that were taken to develop a methodology for obtaining and updating a global biodata inventory with Heil's gold standard reproducibility, a robust codebase, and FAIR data standards. Installation of dependencies and reproduction of the entire analysis can be performed each with a single command, and analysis steps are well documented. All code passes static code checks for formatting, linting, and type compatibility. Much of the code is formally tested with unit and integration tests. Core data products are present in GitHub and archived in Zenodo, with accompanying documentation. While we put a considerable amount of effort into something we believe is reproducible. The first anticipated update will occur in the first quarter of 2023 and will be the first true test. By then most of the research team will have moved on and the remaining members may not be those who directly wrote the code or are programmers themselves. Community interest has been strong so far, but actual reuse is unknown at this time.

The methodologies used in this work are not novel on their own. The automation employed to make reproduction simple relies on the widely used Snakemake workflow manager. Further, it is common practice in software engineering disciplines to leverage static code checking and testing. Regarding data standards, we looked to the FAIR principles. The purpose of this work is

to show how a research project that utilizes computational methods, particularly ML, can be implemented to maintain robustness and strive for a high level of reproducibility. However, we recognize that there are numerous ways to accomplish this and do not mean to claim our implementation is the best.

Certain improvements could be made, such as using a more robust package manager like poetry and using git hooks to automatically run tests upon committing to git (Eustace, 2018). Importantly, test coverage is lacking in some areas, especially for portions that involve heavy computation like model training. Still, the current test coverage is enough to increase confidence in the code's behavior.

Beyond the technical details, it is likely quite apparent that this work required a substantial amount of time to think through and implement. We were able to do this only because of our team's collective belief that these efforts were worth the resources invested. However, not only did it require allocation of those resources, but it also required trust among team members and a willingness to engage in the process. Additionally, many tradeoffs were made. One example is the tradeoff of using only open data versus a more extensive commercial data source, which undoubtedly would have yielded a larger, but in our estimation a less useful, inventory. There are also ambitions that we had at the start of the study that are now future directions because we chose to devote time developing a robust workflow instead. This required principled project management and caused, even as we write this, some amount of wistfulness. In the end, we could not do it "all", and we fully appreciate that others must decide for themselves where to place their efforts.

We hope that this work can provide an example, and hopefully not too daunting of one, to others interested in attempting greater reproducibility in their research. We would be especially gratified to learn of others who answer this work with better solutions than those presented here.

8. Acknowledgements

The authors would like to Ana-Maria Istrate, Donghui Li, and Gully Burns at Chan Zuckerberg Initiative for their fruitful conversations and scientific support; Ken Youens-Clark formerly at The University of Arizona, Alise Ponsero at The University of Helsinki, and Bonnie Hurwitz at The University of Arizona for their mentorship of Kenneth Schackart.

9. Conflict of Interest

The authors declare no conflict of interest.

10. Funding

This work was funded by the Global Biodata Coalition.

11. Code and Data Availability

All code is openly available at https://github.com/globalbiodata/inventory_2022. Additionally, all code and data will be archived in Zenodo when the forthcoming full article describing the study is published.

12. Author Contributions

Project was conceived by CEC; Open Science Implementation plan was developed by HJI and KES; Initial machine learning code was written by AMI and developed into the pipeline described here by KES; Manuscript preparation was performed by AMI, CEC, HJI, and KES. All authors have read and approved the final manuscript.

13. References

Ahmed, H., & Lofstead, J. (2022). Managing Randomness to Enable Reproducible Machine Learning. *Proceedings of the 5th International Workshop on Practical Reproducible Evaluation of Computer Systems*, 15–20. https://doi.org/10.1145/3526062.3536353

anaconda. (2020). Anaconda Software Distribution. *Anaconda Documentation*. https://docs.anaconda.com/

Berman, H., Henrick, K., & Nakamura, H. (2003). Announcing the worldwide Protein Data Bank. *Nature Structural & Molecular Biology*, *10*(12), Article 12. https://doi.org/10.1038/nsb1203-980

Bisong, E. (2019). Google Colaboratory. In E. Bisong (Ed.), *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* (pp. 59–64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7

Cordasco, I., & Ziade, T. (2010). Flake8: Your tool for style guide enforcement. *Programa de Computador*.

de Saint-Exupéry, A. (1943). Le petit prince [The little prince]. *Verenigde State van Amerika: Reynal & Hitchkock (US), Gallimard (FR)*.

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology*, *35*(4), Article 4. https://doi.org/10.1038/nbt.3820

Eustace, S. (2018) Poetry: Python packaging and dependency management made easy. *GitHub.* https://github.com/python-poetry/poetry

Fanelli, D. (2018). Is science really facing a reproducibility crisis, and do we need it to? *Proceedings of the National Academy of Sciences*, *115*(11), 2628–2631. https://doi.org/10.1073/pnas.1708272114

Github (2020). *GitHub*. https://github.com/.

Google (2022) yapf: A formatter for Python files. *GitHub.* https://github.com/google/yapf

Hall, M., & Letcher B., (2020) snakefmt: The uncompromising Snakemake code formatter. *GitHub.* https://github.com/snakemake/snakefmt

Haring, R., & Bell, R. J. (2018). Lack of research reproducibility, the rise of open science and the need for continuing education in research methods. *Climacteric*, *21*(5), 413–414. https://doi.org/10.1080/13697137.2018.1476968

Heil, B. J., Hoffman, M. M., Markowetz, F., Lee, S.-I., Greene, C. S., & Hicks, S. C. (2021). Reproducibility standards for machine learning in the life sciences. *Nature Methods*, *18*(10), Article 10. https://doi.org/10.1038/s41592-021-01256-7Ioannidis, J. P. A. (2005). Why Most Published Research Findings Are False. *PLOS Medicine*, *2*(8), e124. https://doi.org/10.1371/journal.pmed.0020124

Köster, J., & Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19) pg. 2520-2522. https://doi.org/10.1093/bioinformatics/bts480

Krekel, H., Oliveira, B., Pfannschmidt, R., Bruynooghe, F., Laugher, B., & Bruhin, F. (2004). *Pytest 6.2.4*. https://github.com/pytest-dev/pytest

Langa, Ł. (2018) Black: The uncompromising Python code formatter. *GitHub.* https://github.com/psf/black

Lehtosalo, J., Rossum, G. v, Levkivskyi, I., Sullivan, M. J., Fisher, D., Price, G., Lee, M., Seyfer, N., Barton, R., Ilinskiy, S., & others. (2021). Mypy: Optional static typing for Python. http://Mypy-Lang. Org/.

Merkel, D., (2014) Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 239.

Müller, K., & Walthert, L. (2022). *styler: Non-Invasive Pretty Printing of R Code*. https://CRAN.R-project.org/package=styler

Sveidqvist K., (2022) Mermaid: Generation of diagrams like flowcharts or sequence diagrams from text in a similar manner as markdown. *GitHub.* https://github.com/mermaid-js/mermaid/

The Europe PMC Consortium. (2015). Europe PMC: a full-text literature database for the life sciences and platform for innovation. Nucleic Acids Research, 43(D1). https://doi.org/10.1093/nar/gku1061

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1), Article 1. https://doi.org/10.1038/sdata.2016.18

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Rush, A. M. (2020). *HuggingFace's Transformers: State-of-the-art Natural Language Processing* (arXiv:1910.03771). arXiv. https://doi.org/10.48550/arXiv.1910.03771