Fall 2020

Machine Learning Methods for the Analysis of Metagenomes

Vito Adrian Cantu Alessio Robles
Claremont Graduate University

MACHINE LEARNING METHODS FOR THE ANALYSIS OF METAGENOMES

by

Vito Adrian Cantu Alessio Robles

Claremont Graduate University and San Diego State University

2020

Approval of the Dissertation Committee

This dissertation has been duly read, reviewed, and critiqued by the Committee listed below, which hereby approves the manuscript of Vito Adrian Cantu Alessio Robles as fulfilling the scope and quality requirements for meriting the degree of Doctor of Philosophy in Computational Science.

Robert Edwards, Chair
San Diego State University
Professor of Biology

Claudia Rangel
Claremont Graduate University
Adjunct Professor of Mathematics

Anca Segall
San Diego State University
Professor of Biology

Allon Percus
Claremont Graduate University
Professor of Mathematics

Barbara Bailey
San Diego State University
Associate Professor of Mathematics and Statistics

Abstract

Machine Learning Methods for the Analysis of Metagenomes
by
Vito Adrian Cantu Alessio Robles

Claremont Graduate University and San Diego State University: 2020

As of October 2020, there are 18.6×10^{15} DNA base pairs publicly available in the Sequence Read Archive and this number is growing at an exponential rate. As DNA sequencing prices continue to drop, many research groups around the world have incorporated high throughput sequencing in their research, giving us access to sequences from many distinct ecosystems. This has revolutionized the field of metagenomics, which aims to fully characterize all organisms and their interactions in a particular system. Nevertheless, the plethora of available data has made its analysis difficult as traditional techniques such as genome assembly or sequence alignment are bound to fail due to the high noise of metagenomes, or take an impractically long time due to their size. Through this thesis, we explore those challenges and develop techniques to meet them.

Chapter 1 serves as an introduction to the fields of metagenomics and machine learning and the applications where the two meet. Chapter 2 examines the different kinds of noises in sequencing datasets and presents PRINSEQ++, a C++ multi-threaded software for quality control of sequencing datasets. Chapter 3 describes the analysis of 63 metagenomic samples from children with ”nodding syndrome” using Random Forest to give insights into the etiology of the disease. Chapter 4 explores the use of artificial neutral networks to classify phage structural proteins derived from metagenomes.

I wanted to write that my work consists of two parts: of the one which is here, and of everything which I have not written. And precisely this second part is the important one.

– Ludwig Wittgenstein

TABLE OF CONTENTS

PAGE

ABSTRACT iv

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER

1 INTRODUCTION 1

1.1 DNA 1

1.2 DNA sequencing 2

1.2.1 Sanger sequencing 3

1.2.2 Ilumina (solexa) sequencing 3

1.2.3 Oxford Nanopore 4

1.3 Metagenomics 4

1.4 Big data in metagenomics 6

1.5 Machine learning 7

1.6 Training a machine learning model 8

1.6.1 Method selection and data collection 8

1.6.2 Feature extraction 9

1.6.3 Training, Validation and Testing 10

1.7 Machine learning in metagenomics 11

1.7.1 OTU clustering and contig/read binning 11

1.7.2 Taxonomic assignment and diversity profiling 12

1.7.3 Comparative metagenomics 13

1.7.4 Gene prediction and annotation 14

2 PRINSEQ++ 16

2.1 Introduction 16

2.2 Noise in metagenomes 17

2.2.1 Sequence quality 17

2.2.2 Sequence complexity 17

2.2.3 Sequence duplication 18

2.2.4 IUPAC ambiguity code 19

2.3 Parallelization 19

2.3.1 Speedup 20

2.4 QC tools compared 21

2.4.1 Run time 21

2.4.2 Features comparison 22

2.4.3 Memory usage 22

2.5 Code availability 23

2.6 Conclusion 24

3 HALOMONAS ELONGATA AND ITS RELATION TO NODDING SYNDROME 25

3.1 Nodding Syndrome 25

3.2 Nodding Syndrome Metagenomes 26

3.3 Results 28

3.3.1 Random Forest 29

3.3.2 Halophage 32

3.4 Methods 33

3.5 Discussion 35

4 PhANNs 38

4.1 Introduction 38
4.1.1 Phages 39
4.1.2 Artificial neural networks 39
4.2 Paper 40
4.3 Discussion 59
4.3.1 Logistic regression 59
4.3.2 Expanded cluster 59
4.3.3 Model size 59
4.3.4 Web server 61
BIBLIOGRAPHY 63
APPENDICES
A Nodding Syndrome samples 75

LIST OF TABLES

PAGE

1.1 Some machine learning methods 15

2.1 Speedup of multi-threaded PRINSEQ++ 20

2.2 Features of various sequencing QC tools 24

3.1 12 top contigs by t-test p-value. 29

3.2 Annotation of the top 12 contigs 32

3.3 Cross assembly stats 34

4.1 Side chain groups 61

4.2 Model size 62

A.1 NS samples. 76

A.2 Top 100 contig stats 80

LIST OF FIGURES

PAGE

2.1 prinseq-lite and PRINSEQ++ runtime comparison 21

2.2 Run-time comparison of QC tools for fastq files using a single thread. Error bars use a 0.95 confidence interval. 22

2.3 Memory usage comparison of QC tools for fastq files using a single thread. Error bars use a 0.95 confidence interval. 23

3.1 Titule, Democratic Republic of Congo 27

3.2 Boxplots of the top 12 contigs across sample type 31

3.3 Halophage contigs - regions homologous to *halomonas* are shown in blue 33

3.4 Halophage annotation 33

3.5 Proportion of control votes for different RF models 37

4.1 Phage structural proteins 39

4.2 A multi-layer ANN 40

4.3 Comparison of a Logistic Regression, an ANN trained using the reduced set and a ANN trained using the expanded set 60

4.4 “tetra_sc_tri_p” class performance comparison 61

CHAPTER 1
INTRODUCTION

1.1 DNA

Desoxyribonucleic acid (DNA) is an organic molecule composed of two coiled polymeric chains that form a double helix[104]. Each of the two strands is a "Polynucleotide" composed of any combination of four monomeric units called nucleotides. Each nucleotide is made up of one of four nitrogen containing bases (cytosine [C], guanine [G], adenine [A], or thymine [T]), a ribose molecule, and a phosphate group[3]. The DNA double helix is stabilized by hydrogen bonds between the bases of opsite chains that are only formed with the specific pairings A-T and C-G. This fact has two important consequences, two DNA strands will form a double helix only if their nucleotide sequences are complementary, and each strand has all the information needed to deduce the complementary strand. As such, a DNA molecule can be represented naturally by a long string of the characters 'A', 'C', 'T', and 'G'.

Long (thousands to millions of bases) double stranded DNA molecules are called chromosomes. The information contained in the sequences of specific regions of the chromosome, called genes, can be used by the cell as a blueprint, via transcription and translation, to produce proteins that serve as enzymes or structural components. DNA polymerase, one such enzyme, can synthesize the complementary strand of single-stranded DNA molecule making it double-stranded. During cell division, each chromosome is split into its two complementary strands and each strand is replicated by DNA polymerase. This generates two identical chromosomes, one for each of the divided cells.

The DNA double helix has a radius of 2nm ($2 * 10^{-9}$ metre) and a length of 0.34 nm per base. At 3.2 Gbp (Gigabases, or $3.2 * 10^9$ bases), all unique chromosomes of a human cell (humans have two copies of each chromosome) measure about 1 metre long. For other organisms, genome sizes range from a few kilobases (10^3bps) for some viruses, to hundreds of gigabases (10^9 bp) for some plants.

1.2 DNA SEQUENCING

Given the physical size of DNA, and that two distinct DNA molecules have eerily similar chemical properties, elucidating the nucleotide sequence of a particular DNA molecule (aka DNA sequencing) remains challenging despite 40 years of innovation since the first genome, phage MS2 [33] was published in 1976. (Phage ϕ-x174, sometimes credited as the first genome, was published in 1977 [86]). Several techniques and technologies have been developed to sequence DNA. Most of them can only sequence small fragments (reads), anywhere from 35bp for early illumina to a median of 10,000bp for Oxford Nanopore MinION. Furthermore, there is a chance for the technique to misidentify a base (error rate). The process of reconstructing the original DNA sequence from these fragments is called assembly. The naive way to do this is to look for overlapping segments at the ends of each fragment and merge them iteratively to generate large sequences (called contigs). This computationally expensive and modern assemblers use De Bruijn graphs to find these overlaps efficiently [7]. Ambiguous assemblies are produced on repeated regions (either in tandem repeats or with repeats in different parts of the genome) when the repeat is larger than the fragment size. Under ideal conditions, each contig corresponds to a complete chromosome but sequencing errors and repeated regions most often than not causes that several contigs correspond to a chromosome with some unknown regions (gaps) between them. Most sequencing protocols require a few nanograms of DNA. Depending on the sample, this might be hard to obtain. In that case, random polymerase chain reaction (PCR) can be used to amplify (increase the concentration of) DNA in a nonspecific way. If the

flanking sequences for a region of interest are known, PCR can be performed to increase the concentration of DNA from that region, in essence both aplifying and purifying the sample.

1.2.1 Sanger sequencing

This method is named after Frederic Sanger, who in 1980 was awarded the Nobel Prize for the second time ”for their contributions concerning the determination of base sequences in nucleic acids” [1]. Sanger sequencing is one of the first sequencing methods and it is still in use today. In its original form, the target sequence is replicated in four different reactions with the four standard nucleotides. Each reaction is spiked with different di-deoxy-nucleotide labeled with radioactive phosphate. Di-deoxy-nucleotides are inserted in the DNA normally, but prevent further DNA elongation (the addition of nucleotides by DNA polymerase). This generates DNA chains of specific sizes (those whose position corresponds to the spiked di-deoxy-nucleotide base). The DNA sequence can be inferred from the bands of the four reactions in an electrophoresis gel. In its current form, the four di-deoxy-nucleotides are labeled with different fluorescent dyes and the electrophoresis is performed in capillary tubes. Even though the cost per base is high, Sanger sequencing is still in use today as it can sequence long strands of DNA (1,000bp) and a single reaction costs only a few dollars.

1.2.2 Ilumina (solexa) sequencing

Ilumina performs an initial random PCR where adapters with known sequences are added at the start of each DNA fragment, then an additional PCR is performed on the surface of a glass coated with the DNA complementary to the adapter. This produces clonal DNA “clusters” with distinct sequences. Sequencing is performed by cycles, first adding nucleotides modified with different fluorescent dyes according to the base and a removable protective group on the ribose sugar to prevent the addition of more than one base. A high resolution camera determines the position and base added at each cluster. Finally, the protective group and fluorescent dye are removed and the cycle starts anew. Via “bridge amplification” Illumina can sequence up to 300 bp of each end of a DNA molecule. The

highest output equipment can produce up to 20 billion ($20 * 10^9$) sequences or a total of 6 Tbp ($6 * 10^12bp$), in theory enough for a large plant genome. Ilumina is currently the cheapest sequencing technology at just $5 per gigabase. Using tag sequences, up to 384 different samples can be sequenced in a single experiment.

1.2.3 Oxford Nanopore

Nanopore sequencing uses a fundamentally different approach. DNA is put in one of two chambers with a grid of impermeable membrane patches dividing them. Each patch has a nanopore (a protein that forms a hole in the membrane) and an ampmeter that measures the current flowing through the pore. A voltage difference between the two chambers pushes DNA and ions in the buffer through the pore (phosphate groups in DNA are negatively charged). While the current generated by the flow of ions is known and constant, it is disrupted by the DNA in a sequence dependent way. By measuring the current across time for a pore, the sequence of the DNA molecule can be inferred.

Nanopore is the sequencing technology that produces the longest reads (up to 2 millions bases in a single read have been reported [82]) and data is produced in real time as the DNA goes through the pore. Yet its high error rate (5%-25% [107]) and cost ($30-$45 per gigabase) makes it not ideal for large projects. Nanopore is most often used to complement Illumina in resolving assemblies or in time sensitive applications[53].

1.3 METAGENOMICS

Metagenomics is the study of genetic material recovered directly from environmental samples[43]. Forgoing the need for clonal cultures greatly increases the number of organisms that can be studied. Indeed, early metagenomic work [5] (1995) on the diversity of the 16s gene, (a gene common to all bacteria) estimated that more than 99% of bacteria are not cultivable. Furthermore, it has become increasingly evident that knowing the full genome of

an organism helps very little to understand an ecosystem, as bacterial communities are distinct and complex[106].

The limitations of a single genome were made painfully evident when the human genome was published in 2001 [57]. At the time it was stated that “Genetic prediction of individual risks of disease and responsiveness to drugs will reach the medical mainstream in the next decade or so” [20]. Twenty years later, this has yet to happen. Given our close relationship with the microbes in our body, it is now accepted that we will need to gain a deep understanding of our microbiota both in health and disease[77].

There are two type of metagenomic studies, targeted and untargeted. they differ according to the kind of data that is collected. In targeted metagenomics, only partial genomic information is collected. DNA is extracted from all cells in the sample and a set of genes are amplified by PCR, or cloned in artificial chromosomes or plasmids, and only these marker genes are sequenced. The most used marker gene is the small subunit ribosomal RNA (16S in Bacteria and Archaea, 18S in Eukarya) as it is present in all organisms and it is taxonomically informative [78]. The analysis of targeted metagenomic data is simpler as the data produced is less complex and many public databases of marker genes are available [26]. The number and identity of taxa that can be found in a metagenome can vary depending on the marker gene used [38]. Taxa whose target gene is too divergent or absent are excluded from the analysis.

In untargeted metagenomics, the DNA is sequenced without targeted amplification. This allows the identification of new taxa and new genes, the elucidation of the metabolic capacities of the community, and ultimately to gain a better understanding of the ecosystem. Untargeted metagenomic data is significantly harder to analyse as the resulting DNA sequences are heterogeneous, most taxa are new and not found in public databases, and considerably more sequences are needed to draw meaningful observations. Environmental

samples contain anywhere from a few to thousands, or even millions [35, 82] of different taxa. If the community of interest is host associated, care must be taken to exclude its sequences. This can be done by fractional filtration of the sample and removing any leftover host sequences computationally. For example, in human associated samples, without removing the human genetic material it is not uncommon to have 95% or more human sequences [74].

1.4 BIG DATA IN METAGENOMICS

A single metagenomic experiment can produce as much as $6 * 10^{12}$ bases [49]. Inferring relative abundance of species, metabolic pathways, interspecies interactions, and many more useful data from this information is a significant computational challenge, is increased as the number of samples in a study grows or if public metagenomes are used.

As of October 2020 there are $18.6 * 10^{15}$ metagenomic bases (8.6 Petabytes) in the public section of the sequence read archive (SRA), up from $4.5 * 10^{13}$ bases (0.08 Petabytes) just 10 years ago. In the same timeframe the cost of sequencing per gigabase fell from $780 to $8 (97.5 fold reduction)[60] while the computing cost per Gigaflop only fell from $1.80 to $0.03 (60 fold reduction) [109]. From a computational standpoint, the problem with the rapid growth in available data is that most naive approaches to analyze metagenomes run in quadratic time (or worse). If you double the size of the input, the analysis will take four times as long (or you would need a computer four times as fast).

Out of necessity many advanced computational approaches have been used to analyze metagenomes in a manageable timeframe. In this thesis two such approaches have been used to develop tools for metagenome analysis. The first is parallel computing, which will be explored in detail in Chapter 2. The Second is Machine Learning. Specific tools and methodologies using Machine Learning will be explored in chapter 3 and 4. A brief introduction to machine learning follows.

1.5 MACHINE LEARNING

A machine learning algorithm is one whose performance improves with experience. It consists of three parts [70] :

1. Experience, in the form of data.
2. A task, the algorithm that produces an output
3. An objective function, a way to measure the performance of a given output.

This algorithm is said to be "learning" if it's performance for a given task improves as more data is available. In a broader sense, a Machine learning algorithm tries to approximate the real mathematical function (F_{real}) that maps the inputs of a task to its outputs by optimizing the objective function in response to some data. In a sense, the objective function acts as a proxy for the distance between the model and F_{real}.

Some underlying assumptions of all machine learning methods is that all the observed data is generated by F_{real}, that such a function exists, that it (or a close approximation) is computable, and that the method can approximate it. This approximation takes a distinct form (representation) for different ML methods, such as decision trees in random forest, long matrix multiplication in neural networks, or dendrograms in hierarchical clustering. The set of all possible approximations to F_{real} given a representation and the model hyperparameters (parameters that don't change during learning) is called the hypothesis space. The space is structured in such a way that learning is equivalent to a numerical optimization problem to find the hypothesis (F_{real} approximations) whose output maximizes (or minimizes) the objective function. Using more data to generate a model will always result in a model closer to F_{real} [99], this is true even for unsupervised methods [36].

ML methods can be roughly divided into three groups: supervised learning, unsupervised learning, and reinforcement learning. Supervised methods are those where each

data point has a label attached to it. This label can have any form such as a categorical variable, a scalar, a vector or even a tensor. More often than not, the objective of a supervised ML algorithm is to learn the relationship between data and label so that new observations can be labeled. Unsupervised ML methods, on the other hand, don't use labeled data and aim to learn about the underlying structure of the data. Clustering and dimensionality reduction are common examples of unsupervised learning. Finally, a reinforcement ML algorithm "decides" on an action from the input and a probabilistic model to generate and output that is then used as input in an iterative manner. Each iteration gives the model a reward depending on the value of the objective function. Self driving cars are a common application of reinforcement learning.

1.6 TRAINING A MACHINE LEARNING MODEL

Training a machine learning is typically divided in the following three steps:

1.6.1 Method selection and data collection

The ML method to be used is driven by the particular questions to be answered and the training data available. The first thing to consider is whether the ML method can even answer the question. For example, Support Vector Machines can only do a binary classification and linear regression can only approximate linear functions. A more complex model that can approximate nonlinear functions (like Neural Networks) typically has a larger hypothesis space to explore and more parameters to approximate. This means that more training data would be needed. While the exact number or training examples needed depends on the specific problem and how correlated the examples are, a good starting guess is to use 10 examples for every parameter to approximate[41]. For applications where little data is available and generating new data is time consuming and/or expensive, this can preclude the use of some complex ML methods.

In general, you want to use as much training data as you possibly can, with a few caveats. (i) Be aware of any bias in the data. ML models have been known to learn sexist[10] and racist[76] behavior from data. (ii) Be aware of unbalanced data, especially with classification methods. If one of the classes has many more training examples than the others the data is said to be unbalanced. If nothing is done about it, the classifier might learn to always guess the majority class, as it gives a high accuracy. An unbalanced dataset can be addressed by undersampling the majority class, oversampling the minority clases, weighting each class differently during learning and/or artificially generating new training examples (data augmentation[111]). Data augmentation is common in image classification, where slightly rotating or translating the image shouldn't change its label. For applications where data augmentation is feasible, it can also solve the problem of too few training examples per parameter.

While it is important to keep these problems in mind, when comparing the performance of a model trained on a small well-curated dataset vs. a very large uncurated one (several orders of magnitude larger), the model trained on the largest dataset almost always performs better[42].

1.6.2 Feature extraction

ML methods can only use numeric data, both inputs and labels need to be transformed to a numeric vector (or matrix or tensor) of fixed size. This is sometimes a restriction on the method itself and sometimes a restriction on the heuristics used to optimize run time. This can be done in two ways, encoding and feature extraction.

Encoding is representing the data as a tensor. Images are typically represented as a matrix where each entry is the grayscale intensity of a pixel or a 3D tensor of red, blue, and green intensities. Categorical variables can be represented by a "one-hot" encoding, a vector of length equal to the number of categories with 1 in the entry that represents a specific label

and 0 everywhere else. This encoding has the advantage of not assuming any natural ordering between the labels. DNA sequences can be encoded as a matrix of one-hot encoded size 4 columns (one per base) padded or trimmed to a specific row length[62]. Most encodings are reversible, that is the original data can be reconstructed from the encoding.

Feature extraction aims to generate a vector with numbers that represent properties of the data. For example a DNA sequence can be represented by a size 4 vector with the proportion of each base as entries or a node in a network can be represented by a vector with different topological features[98]. The DNA feature vector can be extended to include the length of the sequence, the frequency of dinucleotides, molecular weight etc. The more features you extract from the data the better your method becomes, but more features increases the number of parameters to be estimated and you get diminishing returns, especially with correlated features (in this case the molecular weight can be deducted from the nucleotide frequency and the proportion of each base). Furthermore, features can be transformed by any number of functions. For example, Z-score and sigmoids are common choices as they prevent the features from spanning several orders of magnitude. Most of the time, it is impossible to reconstruct the original data from a feature vector.

The choice of encoding and feature vector can greatly influence the performance of any ML method [72].

1.6.3 Training, Validation and Testing

Training an ML model is equivalent to numerically finding the element of the hypothesis space that optimizes the objective function. Validation is using a metric on the model (which may or may not be the objective function) to measure how well it performs compared to other models. Validation is often used to tune “hyperparameters”. That is parameters that are chosen *a priori* and are not modified during training (e.g. the number of trees in Random Forest or the number of layers and neurons in artificial neural networks).

Training and validation are method specific, but there are some commonalities depending on the type of ML method.

Supervised methods split the data into the training/validation set and a testing set (a 90/10 split is common). The testing set is set aside for the moment. The training/validation set is further split into training and validation sets (again, 90/10 is common). The training set is used to train several models with different hyperparameters, feature vectors, and/or encodings. Then the model is used to predict labels of the validation set. As the real labels are known, the proportion of correctly predicted labels (accuracy) can be computed. The model with the highest accuracy is chosen as the best model. During validation the model is learning from the validation set. To avoid this overtraining bias, the accuracy on the test set, which has never been used to approximate parameters, is also reported.

In unsupervised methods all data is used to train several models with different hyperparameters. The objective function or the "silhouette coefficient"[55] can be used to determine the best model. In reinforcement learning, the cumulative reward over all iterations is used as a validation metric. If the best possible decision is known, the difference between its reward and the reward gained by the algorithm's decision is called "regret". Either the smallest regret or the highest cumulative rewards can be used to select the best model.

1.7 MACHINE LEARNING IN METAGENOMICS

Many machine learning methods have been implemented to interrogate metagenomic samples (see Table 1.1). Machine learning is currently used in metagenomic analysis to answer the following questions:

1.7.1 OTU clustering and contig/read binning

Targeted metagenomics (amplicon sequencing) is a popular way to explore the genetic diversity of a sample by clustering sequences into Operational Taxonomic Units (OTU). OTUs are groups of closely related sequences. Depending on how close you require them to be, OTUs can be a proxy for species, genus or other taxonomic level. The 16S gene is the most used marker for OTU clustering [65], but any conserved gene should work. Contig (or read) binning is the analogous question for untargeted metagenomics. Both OTU clustering and contig binning typically require a distance function d and a threshold t under which two sequences are considered close enough. In the case of OTU clustering, alignment score (or sequence similarity) is a natural choice for distance. For contig binning, the choice of distance is less obvious. A combination of k-mer distribution and contig abundance (coverage) is often used as it has been observed that the distribution of k-mer compositions is stable across a single genome and varies between genomes[54].

No matter the metric used, computing the pairwise distance of all OTUs or contigs is slow and escalates poorly as more sequences are considered. Heuristic greedy clustering saves time by first sorting the sequences by some meaningful criteria (e.g. CD-HIT[59] uses length), computes the distance of the fist sequence to all other sequences and forms a cluster out of itself and all sequences t or closer and removes them from the list. This process is repeated for the new first sequence on the list until all sequences are in a cluster.

1.7.2 Taxonomic assignment and diversity profiling

While OTU clustering and contig binning give us groups of sequences that represent a single taxa it doesn't tell us which taxa. Taxonomic assignment aims to label each sequence or cluster with the most specific label from the taxonomic hierarchy, clearly a supervised ML problem. Traditionally, this has been done using non ML methods such as sequence identity to large databases[8]. As both public databases and metagenomic studies increase in size, the

length of time taken for the database comparisons prevent those methods from being practical because the complexity is $O(m \times n)$ where m =number of sequences in database and n is number of sequences in sample. Once a ML model is trained, labels can be assigned to new observations in linear time $O(n)$. The features used for this classification are nucleotide *k*-mers of different sizes[100], but Laplace smoothed *k*-mer counts[79], CG content[16] and hidden Markov model alignments[23] have also been used.

Diversity profiling aims to elucidate the proportion of each taxa in a sample. While this proportion can be easily derived from the taxonomic assignment of reads, contigs or OTUs, some approaches forgo this intermediate step to gain speed. For example FOCUS[90] uses non-negative least squares on the *k*-mer frequencies of known taxa to infer which proportion of those would have given rise to the observed *k*-mer frequency in a metagenome.

1.7.3 Comparative metagenomics

Comparative metagenomics interrogates full metagenomic samples as they relate to other samples. Supervised methods assign a label to each sample, instead of to the contigs or OTUs in it. *K*-mer composition and their correlations[27], OTUs frequencies[95], and contig coverage (see chapter 4) have all been used as features. An active area of research is to use comparative metagenomics as diagnostics. For instance, DectICO[27] has been used to classify irritable bowel disease and asthma samples. Furthermore, some ML methods such as Random Forests can interrogate the features for their contribution to the classification. If the features used are contigs or OTUs this provides an indication of which organisms or genes are associated with the disease (more on this on Chapter 3)

Unsupervised methods cluster metagenomes by similarity. Some distances used are the Unifrac distance[63] (the fraction of taxa present in only one sample, weighted by the length of its branch) for OTUs or the Jaccard distance between the *k*-mer sets of the two

metagenomes[50]. If the metagenomes were cross-assembled [29], the Euclidean distance between the vectors of contig hits per sample can be used.

1.7.4 Gene prediction and annotation

Most protein coding genes contain common DNA sequence features; they start with the 3-mer (codon) ATG, followed by some number of non overlapping 3-mers and end with a stop codon (usually TAA, TAG, TGA). Any section of DNA with these properties is called an Open Reading Frame (ORF) but not all ORFs are translated into proteins. Gene prediction ML aims to learn properties of protein coding ORFs from a database of known genes and known non coding ORFs so that new ORF in a metagenomic sample can be labeled. Common features used are ORF length, CG content, *k*-mer profiles, and distance between contiguous ORFs[45, 75, 84, 112].

Gene annotation is the process of elucidating the biological function of a gene. Doing this experimentally is so expensive and time consuming that it has only been done for a handful of genes[66]. Computational gene annotation has traditionally been done by sequence identity to genes of known function, the assumption being that two genes that are similar are homologous (share a common ancestry) and thus have the same function. As databases and metagenomic experiments become larger, this approach becomes intractable. Another limitation is that due to convergent evolution, two proteins that share functions are not necessarily homologous. PhANNs[14] uses amino acid *k*-mer profiles and some biochemical functions to train an artificial neural network to annotate phage ORFs (more on this on chapter 4).

Table 1.1. Some machine learning methods

Algorithm	type	output	Hypothesis space representation	Metagenomics application	Example tool
K-nn	supervised	categorical	Labeled partition of feature space	Binning, OTU clustering	DOTUR[87]
Support Vector Machines	supervised	scalar	Labeled partition of feature space	Gene prediction or annotation, comparative MG	MetaDistance[61], PVP-SVM[67], DectICO[27]
Linear regression	supervised	scalar	Linear model	binning	Tetra[96]
Logistic regression	supervised	categorical	Linear model	Gene prediction or annotation, comparative MG	MetaGene[75]
Non negative least square	supervised	vector	Vector space	Diversity profiling	FOCUS [90]
Random Forest	supervised	categorical	Decision trees	Taxonomic assignment, comparative MG, gene annotation	16S classifier[16]
Neural Network	supervised	Scalar, tensor or categorical	Neural networks	Gene prediction and annotation	Orfelia[45], PhANNs[14]
PCA	unsupervised	vector	Linear combination of features	binning	CONCOCT[4]
k-means	unsupervised	categorical	means	binning	Metacluster[103]
Hierarchical clustering	unsupervised	categorical	dendrogram	OTU clustering	ESPRIT[93]
Hidden Morkow model	supervised	categorical	network	Gene prediction	FragGene Scan[84], MetaGeneMark[112]

CHAPTER 2
PRINSEQ++

In this chapter we examine the different sources of noise in sequencing datasets and present PRINSEQ++, a C++ multi-threaded software for quality control of sequencing datasets. We also measure PRINSEQ++ speed against other commonly used QC tools.

2.1 INTRODUCTION

As DNA sequencing prices fall, high-throughput sequencing is being used in new and creative ways and areas such as personalized medicine [110] and recreational genomics [32]. This brings about novel challenges for the techniques we use to analyze and draw conclusions from sequencing data, in particular speed and scalability.

Quality control is a crucial step in the analysis of sequencing datasets as low-quality sequences, sequence contamination, and artifacts can eventually lead to erroneous conclusions. Most applications for quality control and preprocessing are written in high level programming languages such as Perl (prinseq-lite [88]) or Java (fastQC [6]) which are slower to execute and provide limited multi-threading support.

Since its publication in in early 2011, prinseq-lite has been cited more than 1,500 times and downloaded more than 54,000 times. In the same time interval, the number of bases in the Sequence Read Archive has grown 617x (from 74 Tbp to 45,704 Tbp) while the computing cost per Gigaflop only fell from $1.80 to $0.03 (60 fold reduction) [109] . It is clear that a new tool, one that has the usefulness of prinseq-lite while being drastically faster, is needed.

PRINSEQ++ implements all the functionality of the Prinseq-lite tool, adds some new features, but can run 16x times faster as it is written in C++ and can take advantage of multi-threading.

2.2 NOISE IN METAGENOMES

No sequencing experiment is 100% accurate. From the different affinity of the polymerise for distinct nucleotides to small errors in manufacturing, noise is always present in sequencing datasets. Systemic errors produce noise that can be mitigated once the processes that generate them are understood. We explore some common sources of noise and methods to reduce it.

2.2.1 Sequence quality

During a sequencing run, each base is assigned a quality score (Q), ranging from 1 to 40, representing the error probability (p). $Q = 10 \Rightarrow p = 0.1$, $Q = 20 \Rightarrow p = 0.01$ and in general $Q = -10log_{10}p$. A Q of 30 ($p = 0.001$) is acceptable for Illumina bases and Q=20 is acceptable for Nanopore bases. The quality tends to degrade closer to the 3' end of the read.

PRINSEQ++ allows you to remove reads with low mean quality score, reads where any base is lower quality than a set threshold and/or to trim the read 3' (or 5') end until a desired mean quality score is reached.

2.2.2 Sequence complexity

Sequence complexity measures how much a particular sequence of symbols differs from a random sequence of symbols using the the same alphabet. Complexity has been used to elucidate if an undeciphered script is written in a real language[83]. In the context of DNA, you would expect close the random (about 90% of the max shannon's index [85]) distribution of *3*-mers. Low sequence complexity, whether biological or generated by the sequencer is commonly non informative and can severely affect the quality of a subsequent genome

assembly. PRINSEQ++ allow you to remove sequences with complexity under some cutoff value.

Total sequence complexity is evaluated as the mean complexity of windows of size 64 or less and step size 32 over the whole sequence. Two methods are available in PRINSEQ++, block-entropies (Shannon-Wiener) and DUST[71] (used by the *blast* algorithm to mask low complexity regions). Both scores are scaled to the [0,1] interval to make the score valid for sequences of any length. In each window, the deviation of the actual counts of each trinuclotide to the expected counts is computed as:

For DUST:

$$CD = \sum_{i=1}^{K} \frac{n_i(n_i - 1)(w - 2)s}{2(l - 1)l} \tag{2.1}$$

For block entropy:

$$CE = -\sum_{i=1}^{K} \frac{n_i}{l} log_k \left(\frac{n_i}{l}\right) \tag{2.2}$$

Where K is the size of the set of all tri-peptides ($4^3 = 64$), n_i is the counts of the ith trinucleotide, w is the window size(64 except at the end of the sequence), l is the the number of trinucleotides in the window (62 for a window of size 64) and s is the scaling factor 1/30

2.2.3 Sequence duplication

Sequence duplications may occur at different steps of the sequencing protocol [37]. Traditionally, duplicated sequences are hard to detect as the naive approach is to compare every single sequence to every other. This is problematic in a multi-threaded environment were each thread holds in memory a few sequences at most and cross talk between threads needs to be minimum in the interest of speed.

PRINSEQ++ uses a probabilistic data structure, a Bloom filter [9][80], to identify duplicate sequences. A Bloom filter is bit-array where every sequence is transformed by several fast (non-cryptographic) hash functions into a bit-array of the same size and the results are operated by a bit-wise OR. To see if a sequence is already in the filter (if it is duplicated) one only need to check the corresponding bits. Bloom filters have false negative rate of 0 (you can be sure that an sequence is not in the structure), and a false positive rate of $\approx \left(1 - e^{-kn/m}\right)^k$ where k is the number of hash functions, n is the number of elements in the structure and m is the number of bits in the bit array. Reading and writing from a bloom filter is fast and can be done asynchronously.

2.2.4 IUPAC ambiguity code

The International Union of Pure and Applied Chemistry (IUPAC) define ambiguity codes for nucleotides whose identity is not fully known [51]. The most common one in metagenomes is N which represent an unknown nucleotide. A large number of Ns in a sequence might indicate low quality or short sequences. Furthermore, some downstream analysis tools (such as those that encode DNA in 2-bit characters) have trouble dealing with Ns and are either converted to As or a random base. PRINSEQ++ allows you to filter out read that have high count of N bases, either by percentage or by absolute number.

2.3 PARALLELIZATION

Most parallelization models, like OpenMP, MPI or Cuda, require the user to know the size and shape of the input *a priori*. Counting the number of sequences in a FASTA or FASTQ file requires reading it completely, which is slow and non trivial (especially so for compressed files). PRINSEQ++ uses POSIX threads (*pthreads*, an application programming interface (API) designed to allow maximum freedom to developers of multi-threaded applications.

With the exception of sequence duplication, the quality control and preprocesing is independent for each sequence pair. Each thread performs all necessary operations on one

sequence pair at the time. This includes: reading from file, uncompressing if necessary, checking for duplicates and other filters, compressing if necessary, and writing to the corresponding output file if the read pair passes all filters. This model drastically reduces run-time, is input size agnostic, and uses little memory.

2.3.1 Speedup

To assess the effect of increasing the number of threads on speed and the speedup of PRINSEQ++ over prinseq-lite, we measured run-time of prinseq-lite and PRINSEQ++ on several FASTQ pair files of different sizes. A pair of FASTQ files from a metagenomic sample were downloaded from the sequence read archive (Run:SRR7091319). The FASTQ files were cut into files of 1, 2, 3, 4, 5, 10, 15, 20, 25, 30 million read pairs. PRINSEQ++ and prinseq-lite were run on those files with equivalent filtering options ("`min_len 100 -min_gc 40 -max_gc 60 -lc_method entropy -lc_threshold 90`" for prinseq-lite and "-min_len 100 -min_gc 40 -max_gc 60 -lc_entropy=0.9" for PRINSEQ++).

Run-time was measured using GNU time 1.7 on a 24 cores Intel Xeon CPU X5650 running at 2.67GHz with 189Gb of RAM. Each measurement was done three times and the mean time and 0.95 confidence interval were plotted on Figure 2.1. Table 2.1 shows the speedup of multi-threaded PRINSEQ++ over prinseq-lite and over single-threaded PRINSEQ++. It is noteworthy that even using a single thread, PRINSEQ++ is about twice as fast as prinseq-lite. This speed gain arose mainly from the switch from Perl to C++.

Table 2.1. Speedup of multi-threaded PRINSEQ++

Threads	speedup over prinseq-lite	speedup over PRINSEQ++ (1 thread)
1	1.98 x	1 x
2	3.77 x	1.93 x
4	7.26 x	3.7 x
8	12.96 x	6.62 x
16	16.47 x	8.39 x

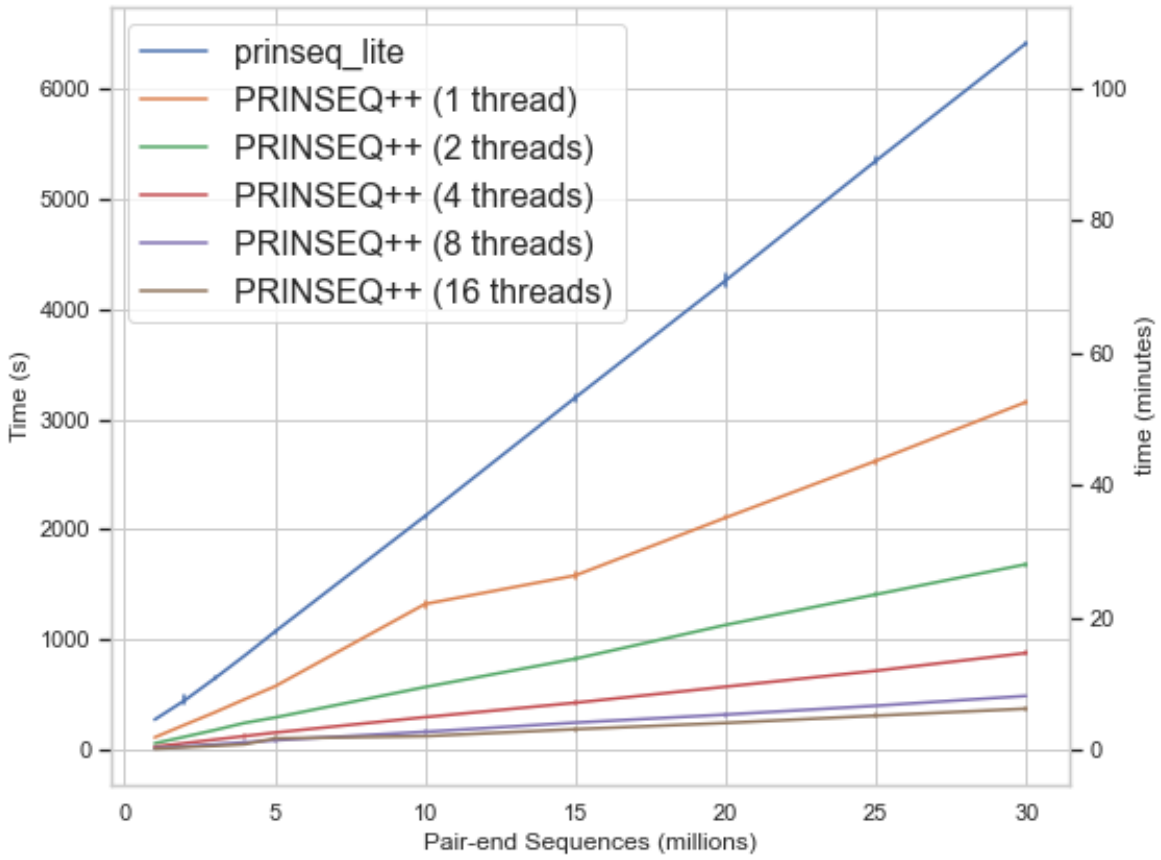


Figure 2.1. prinseq-lite and PRINSEQ++ runtime comparison

There are two main reasons why the speedup does not scale linearly with the number of threads for PRINSEQ++. Input and output files need to be accessed synchronously There is a small overhead in creating a thread. A thread cannot write or read if another thread is doing so, and must wait for it to finish. As the number of threads increases this happens more often and more time is spent waiting for access to files. Additionally, there is little advantage in using more threads than the number of cores in the CPU, as this will cause multiple threads to run on the same core and share execution cycles.

2.4 QC TOOLS COMPARED

In the section we compare PRINSEQ++ speed and performance against other commonly used QC tools.

2.4.1 Run time

Programs were restricted to a single thread with a minimum sequence length of 100 base pairs. Additionally, 15 base pairs at each end were trimmed. The time trials were run in triplicate for each size fastq file.

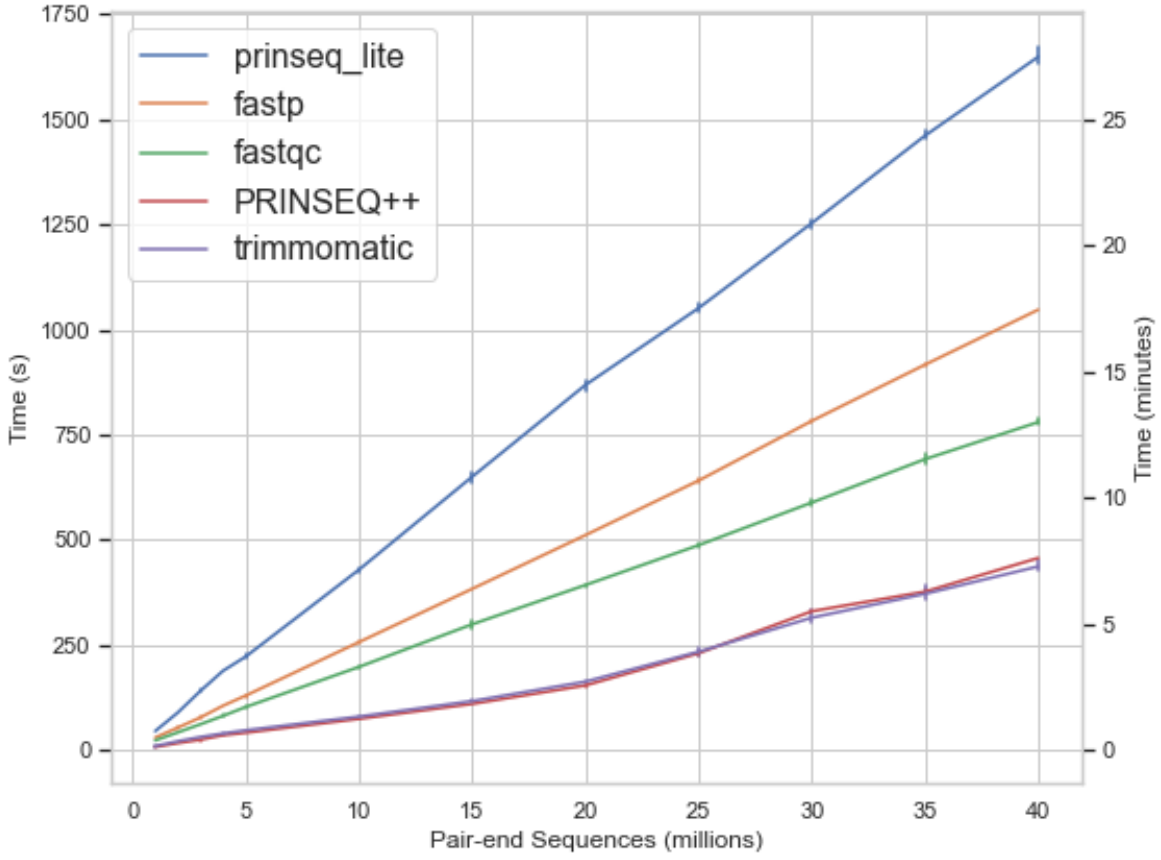


Figure 2.2. Run-time comparison of QC tools for fastq files using a single thread. Error bars use a 0.95 confidence interval.

2.4.2 Features comparison

As no two QC tools have the same implementation and features, a comparison based solely on speed is unfair and uninformative. Table 2.2 shows a comparison of the features implemented for each software tool.

2.4.3 Memory usage

For all software compared, memory usage is not affected that much by the number of sequences in the input file as only a few sequences need to be loaded in memory at the time. How many sequences are loaded in memory, what extra information is needed on how efficiently it is represented determine the memory usage. PRINSEQ++ only store two sequence and the bloom filter in memory, using only a few Mbytes (see Figure 2.3).

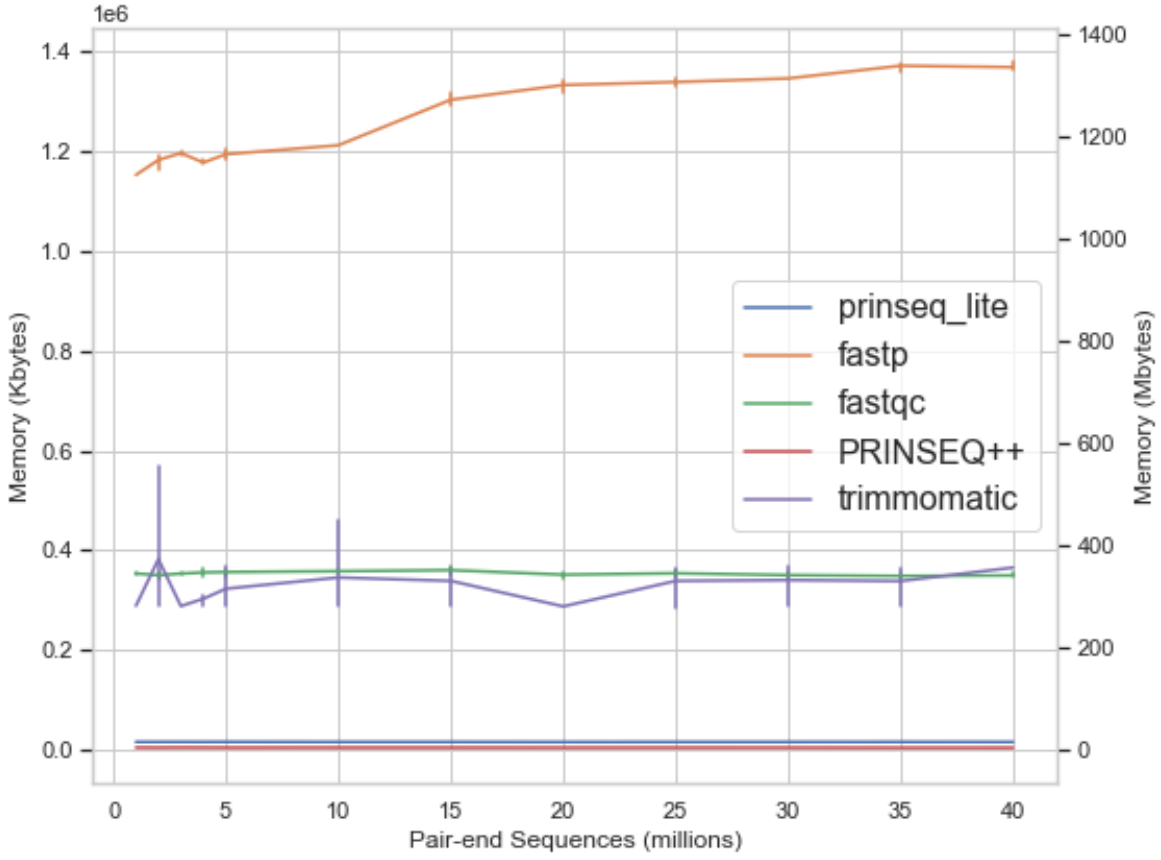


Figure 2.3. Memory usage comparison of QC tools for fastq files using a single thread. Error bars use a 0.95 confidence interval.

Table 2.2. Features of various sequencing QC tools

	Multi-threaded	Fasta Input	Fastq Input	Length Filter	CG% Filter	Quality Filter	Trimming	Compressed Files IO
PRINSEQ++	✓	✓	✓	✓	✓	✓	✓	✓
prinseq-lite		✓	✓	✓	✓	✓	✓	
fastp	✓	✓	✓	✓		✓	✓	Output only
fastQC			✓					Output only
trimmomatic	✓		✓	✓		✓	✓	Input only

2.5 CODE AVAILABILITY

PRINSEQ++ code and binaries are available in GitHub https://github.com/Adrian-Cantu/PRINSEQ-plus-plus and can be built using GNU Autotools or Conda.

2.6 CONCLUSION

PRINSEQ++ is fast and efficient and can significantly reduce the run-time of sequencing datasets analysis. This is critical in applications that are time-sensitive or where the amount of data is so large that slower method are not feasible. PRINSEQ++ has the capacity or reading from, and writing to compressed files without ever uncompressing the whole file, this drastically reduces use hard-drive use. PRINSEQ++ emulates prinseq-lite syntax, thous it can be easily added to any pipeline currently using prinseq-lite.

CHAPTER 3
HALOMONAS ELONGATA AND ITS RELATION TO NODDING SYNDROME

3.1 NODDING SYNDROME

Nodding syndrome (NS) is a neuropsychiatric and epileptiform disorder of unknown etiology that primarily affects children under fifteen years old. The disease is characterized by stunted growth, neurological deterioration, and the eponymous head-nodding epileptic seizures[48]. Nodding syndrome was first reported in Tanzania in 1965,[2] with subsequent reports in Liberia[102], South Sudan[91], Uganda [89] and the Democratic Republic of the Congo[17]. Many possible causes for nodding syndrome have been proposed, including prions[48], mercury exposure[34], or genetic factors[28] but they each have weak correlations with nodding syndrome cases. An alternative hypothesis is that nodding syndrome is related to infection with the filarial parasite *Onchocerca volvulus*, the cause of onchocerciasis and river blindness. Female blackflies (genus *Simulium*) transmit *O. volvulus*, and cases of nodding syndrome occur where both the blackflies and *O. volvulus* are endemic. The increased application of ivermectin (which kills the microfilaria but not the adult *O. volvulus*) to reduce the incidence of onchocerciasis also resulted in a localized reduction in NS cases, suggesting that *O. volvulus* is involved in both diseases [18]. However, *O. volvulus* does not cross the blood-brain-barrier and there is no evidence of *O. volvulus* in cerebrospinal fluid (CSF), suggesting that the connection between *O. volvulus* and NS are indirect. A high-throughput proteomic screen identified antibodies in NS patients that react with human proteins. This lead to the hypothesis that onchocerciasis results in cross-reacting autoimmune antibodies[52]. In this model, patients infected with *O. volvulus* produce antibodies against the *Onchocerca volvulus'* tropomyosin protein that react with the Human leiomodin-1 protein

in the brain causing neurotoxicity. Anti-leiomodin-1 antibodies were detected in the CSF of NS patients and were cross-reactive to *O. volvulus*. However, leiomodin-1 antibodies were also present in almost one-third of unaffected control cases from the same village. Moreover, neurological damage may trigger higher levels of antibodies in the bloodstream obfuscating cause and effect [101].

Like many other parasites, *Onchocerca volvulus* harbors the intracellular endosymbiont *Wolbachia* [56]. The levels of *Wolbachia* in the filarial parasite correlate with disease outcome, suggesting that the endosymbiont contributes to virulence of the nematode[44] . However, *Wolbachia* endosymbionts are essential for *Oncherca*, contributing to their metabolism and reproductive success [47, 94]. Tetracyclines can be used to reduce *Wolbachia* load in the nematode, resulting in sterilization or death of *O. volvulus*. Doxycycline has been used previously to treat onchocerciasis [25] and it is currently being used in a clinical trial to investigate whether it impacts the frequency of epileptic episodes, microfilarial mass, and autoimmune antibodies [101, 47].

For an unbiased exploration of the causes of nodding syndrome, and to explore the possibility of a viral connection with Nodding Syndrome, we extracted viral DNA from the plasma, buffy coat, and cerebrospinal fluid (CSF) of eighteen nodding syndrome patients, as well as nine plasma samples of healthy children from the town of Titule, Bas-Uélé Province of the Democratic Republic of Congo (Figure 3.1). We were unable to identify any eukaryotic viruses uniquely associated with NS, nor identify any *Wolbachia-like* sequences in these samples (but neither have been found in CSF previously). We identified a few sequences that were similar to *Onchocerca flexuosa*, but none that were similar to *O. vulvulus*. However, our novel computational analysis of the sequences revealed a correlation between nodding syndrome and a virus (phage) that infects *Halomonas*-like bacteria that we hypothesize is associated with *Onchocerca volvulus* and is associated with the disease.

3.2 Nodding Syndrome Metagenomes

A case control study was previously conducted in Titule, Bas-Uélé Province of the Democratic Republic of Congo (Figure 3.1) in June 2014 to determine the biological correlates of nodding syndrome. This study was supported by the local health program “Relais Communautaire”, led by the Head Doctor and a team of volunteer members. The Relais Communautaire was a community surveillance network involved in the prophylaxis and treatment of onchocerciasis using Ivermectin. Titule is holoendemic for onchocerciasis with a very high prevalence of epilepsies, the common clinical presentation of NS. This case control study revealed that *Onchocerca volvulus* DNA was detected by PCR in 26/34 (76%) of cases and 10/14; 71% controls[19]. A subset of these samples was used for metagenomic analysis.

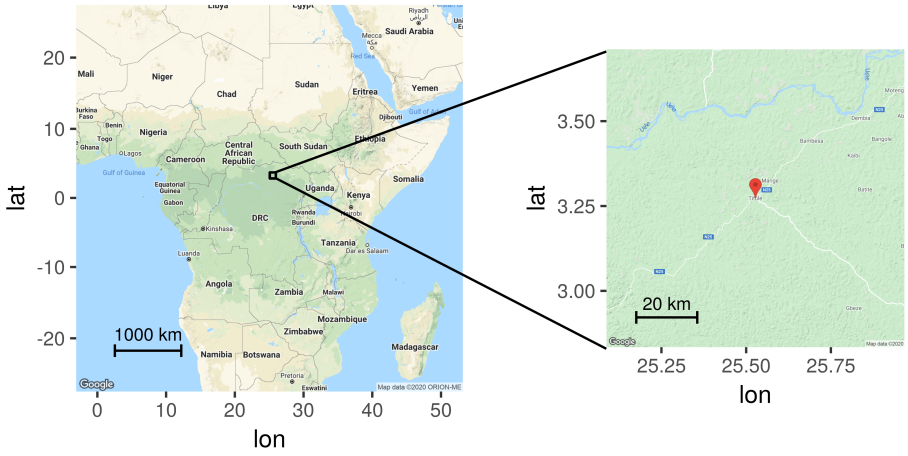


Figure 3.1. Titule, Democratic Republic of Congo

Eighteen individuals with NS were enrolled as cases[19]. Case definitions included a history of at least 2 episodes of unprovoked generalized tonic seizures and absence of known etiology. Nine healthy individuals with no clinical symptoms, that lived in the same or nearby villages and who did not belong to a family with cases of epilepsy were recruited as controls. A written informed consent was obtained from each participant in his/her native language by physicians using a standardized questionnaire. After the interview, cases and controls were examined by a physician. Lumbar punctures were performed by a physician, who had

received special training in neurology while working as a medical doctor in a trypanosomiasis treatment program. After the procedure patients were able to rest and received paracetamol. Blood samples were collected from all cases and controls in heparinized collection vials.

Samples were processed as previously described [15, 24]. Briefly, 110 µl of cerebrospinal fluid, plasma or buffy coat collected from nodding syndrome patients and controls were spun down to remove cells and 100 µl of the supernatant was subjected to DNase treatment to eliminate background cellular DNA with 20 U TURBO™DNase (Ambion). Nucleic acids were extracted from the pre-treated samples as described by Boom [11]. In order to subsequently detect RNA viruses a reverse transcription with 200 U of Superscript II (Invitrogen) and non-ribosomal hexamers [31] was performed followed by a second strand synthesis with 5 U of Klenow fragment (3'-5' exo-) (New England Biolabs) and 7.5 U of RNase H (New England Biolabs). Samples were purified by a phenol chloroform extraction and ethanol precipitation. Subsequent Illumina MiSeq library prep on the dsDNA was performed as described [21]. This treatment should greatly reduce the concentration of Human DNA, leaving mostly bacterial and viral DNA. A total 63 libraries (18 NS plasma, 18 NS buffy coat, 18 NS CSF, and 9 control plasma) 2 x 250bp were sequenced at the Sanger institute (UK) and have been publicly available in the SRA (https://www.ncbi.nlm.nih.gov/bioproject/PRJEB9580 since September 9, 2015).

3.3 Results

To perform an unbiased identification of sequences that are enriched in nodding syndrome, sequences were assembled into contigs and machine learning was used to classify the contigs. All metagenome reads were assembled together (Cross Assembly) and 27,341 (82.5%) of the 33,142 contigs larger than 800 nt (see Table 3.3) were assigned a taxonomic annotation[46]. Twenty-two contigs were identified as coming from *Onchocerca flexuosa*, though none were identified as *O. volvulus* .

The metagenome reads were mapped to the assembled contigs to generate a matrix (the contig/hits table) that indicates how many times a contig is observed in each sample . An unpaired t-test was used to compare normalized contig/hits for the nodding syndrome samples (n=54) and controls (n=9). 132 contigs were significantly different between the two groups (p-value $\leq 10^{-6}$). Of those contigs, 65 were identified as being from the genus *Halomonas* and an additional 11 were from the family Halomonadaceae but the genus could not be identified. Table 3.1 shows the details for the 12 top contigs by t-test p-value. Those results strongly suggest that a member of the *Halomonas* genus co-occurs with NS.

Table 3.1. 12 top contigs by t-test p-value.

contig	annotation	genra	family	$-log_{10}(p\text{-}value)$
NODE_180	Halomonas beimenensis	Halomonas	Halomonadaceae	10.04
NODE_588	Halomonas sp. HG01	Halomonas	Halomonadaceae	10.0
NODE_382	Halomonas sp. 1513	Halomonas	Halomonadaceae	9.89
NODE_738	Halomonas sp. 1513	Halomonas	Halomonadaceae	9.79
NODE_216	Halomonadaceae	Chromohalobacter	Halomonadaceae	9.77
NODE_438	Halomonadaceae	NA	Halomonadaceae	9.70
NODE_578	Chromobacteriaceae	Pseudogulbenkiania	Chromobacteriaceae	9.657
NODE_449	Halomonas sp. HG01	Halomonas	Halomonadaceae	9.52
NODE_183	Halomonas aestuarii	Halomonas	Halomonadaceae	9.50
NODE_269	Halomonas sp. HG01	Halomonas	Halomonadaceae	9.43
NODE_302	Halomonas	Halomonas	Halomonadaceae	9.26
NODE_146	Halomonas sp. 1513	Halomonas	Halomonadaceae	9.19

3.3.1 Random Forest

A Random Forest model was trained to distinguish between Nodding Syndrome samples and controls. Random Forest is a machine learning ensemble method for classification that works by re-sampling the metagenomes and constructing decision trees using the hits to contigs for each sample. Since the metagenomes are re-sampled with replacement, in each iteration some metagenomes may be sampled twice or more, while other metagenomes will not be sampled (This method of re-sampling is also known as

bootstrapping). Each metagenome is, on average, left out from one third of the subsampled sets, however, by repeating the sampling and building the decision trees many times, the entire data set is analyzed. Each tree in the random forest votes to classify a metagenome as either a nodding syndrome case or a control. The out-of-bag (OOB) error is the proportion of misclassified left-out samples when using only the trees where they were left out. The OOB error is a good estimator of the generalization error [12] and can be used to assess the quality of the model. Our final RF model has a OOB error of 1.59% (one misclassification, see Figure 3.5-C).

From this model, the importance of each contig to the classification can be measured by permuting the values for each row of the hit/contig matrix one at time and classifying the modified columns. The OOB error will increase if the permuted contig was important for the classification. The importance of a contig is expressed as the mean decrease accuracy, that is, the average number of extra misclassified metagenomes per tree. We trained two random forest models: The first using all contigs (figure 3.5-A), the second, using the 100 most important contigs in the first model(Figure 3.5B). At a family level, from the 100 most important contigs 70 were annotated as "*Halomonadaceae*". At the genus level, 58 were annotated as "*Halomonas*" (see Table A.2). From the most important 12 contigs from the second RF, 11 are "Halomonas" (see Table 3.2, Figure 3.2). This strengthens the hypothesis that a member of the Halomonas genus co-occurs with NS. The identity of those contigs to the closest halomonas reference is around 80%.

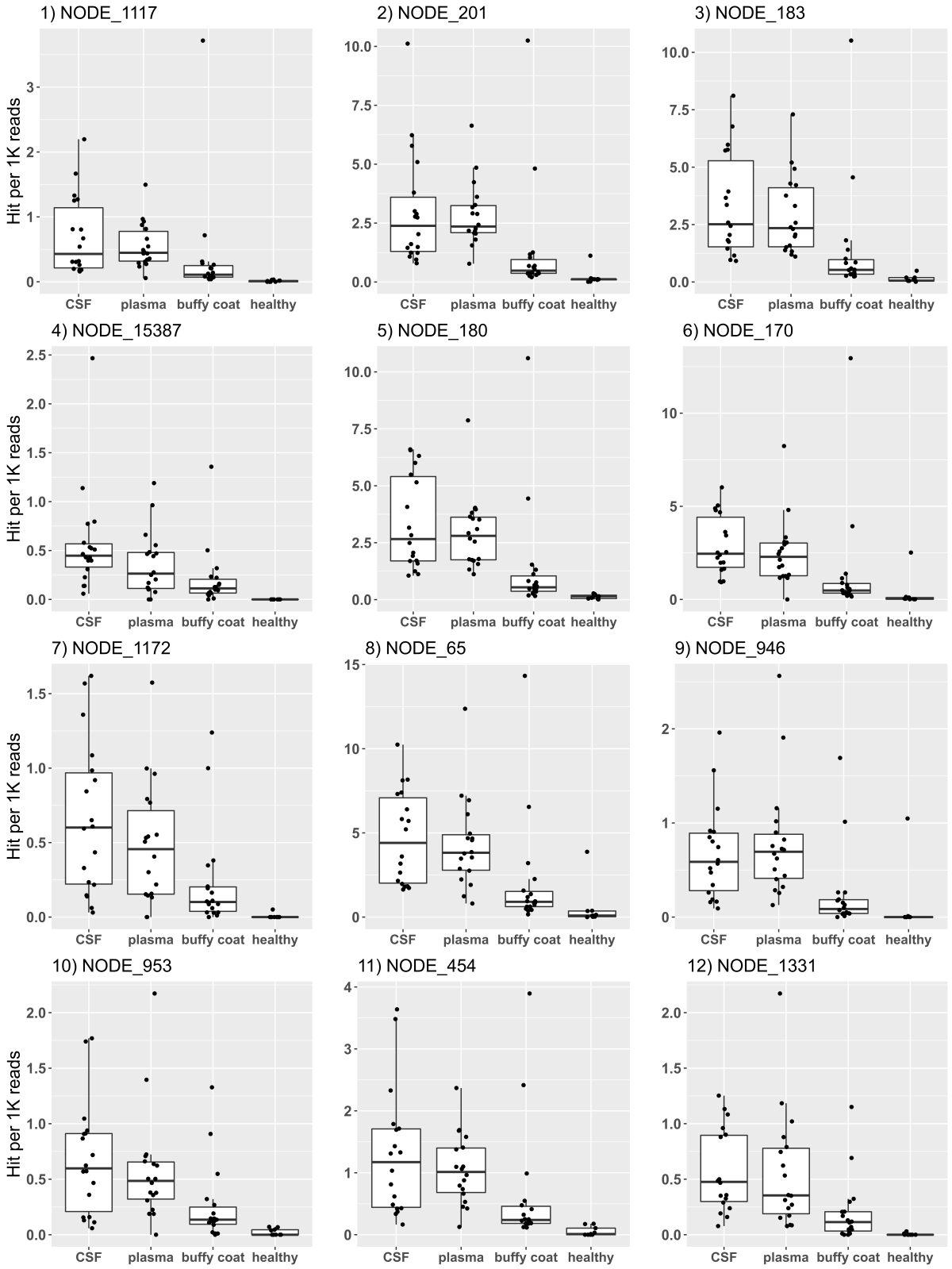


Figure 3.2. Boxplots of the top 12 contigs across sample type

Table 3.2. Annotation of the top 12 contigs

contig	annotation	genus	family	importance
NODE_1117	Halomonas	Halomonas	Halomonadaceae	0.0295
NODE_201	Halomonas sp. HG01	Halomonas	Halomonadaceae	0.0090
NODE_183	Halomonas aestuarii	Halomonas	Halomonadaceae	0.0082
NODE_15387	No hits	NA	NA	0.0074
NODE_180	Halomonas beimenensis	Halomonas	Halomonadaceae	0.0067
NODE_170	Halomonas sp. 1513	Halomonas	Halomonadaceae	0.0053
NODE_1172	Halomonas sp. 1513	Halomonas	Halomonadaceae	0.0052
NODE_65	Halomonas sp. 1513	Halomonas	Halomonadaceae	0.0044
NODE_946	Halomonas	Halomonas	Halomonadaceae	0.0041
NODE_953	Halomonas beimenensis	Halomonas	Halomonadaceae	0.0040
NODE_454	Halomonas sp. 1513	Halomonas	Halomonadaceae	0.0038
NODE_1331	Halomonas sp. 1513	Halomonas	Halomonadaceae	0.0036

3.3.2 Halophage

Manual exploration of the assembly graph around *Halomonas* contigs with high importance reveals that an *Halomonas* contig (NODE_705) along with other non *Halomonas* contigs form a circular DNA structure 57 kbp in length (see Figure 3.3). Furthermore, NODE_705 and NODE_81 (another *Halomonas* contig) share a 56 bp region which has sequencing depth roughly equal to the sum of adjacent regions in NODE_705 and NODE_81 . This suggests that this 57 kbp region is a phage, and that it is found both as a circular molecule and inserted in the *Halomonas* genome in our samples. We named this phage "Halophage". We looked for the function of ORFs in the Halophage using various tools (see Figure 3.4) on identify some clearly phage genes such as an integrase and several phage tails.

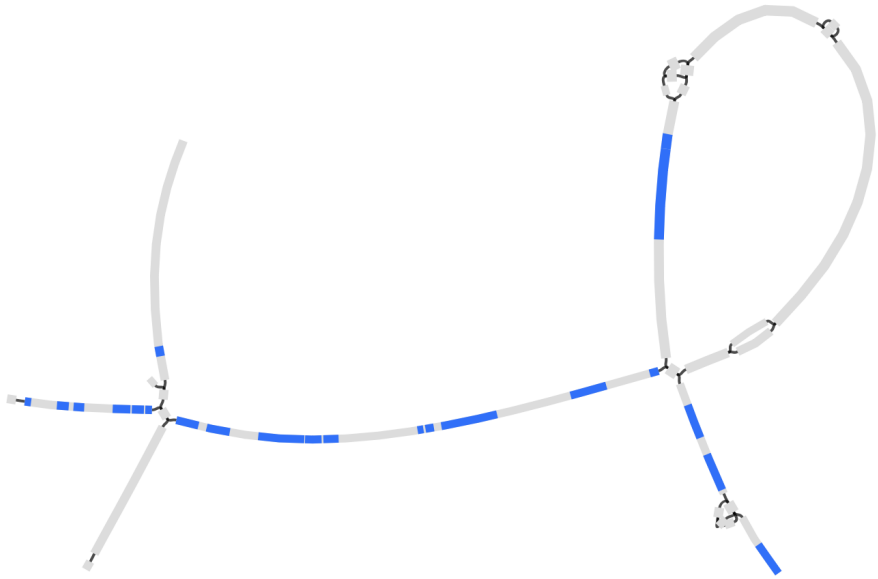


Figure 3.3. Halophage contigs - regions homologous to *halomonas* are shown in blue

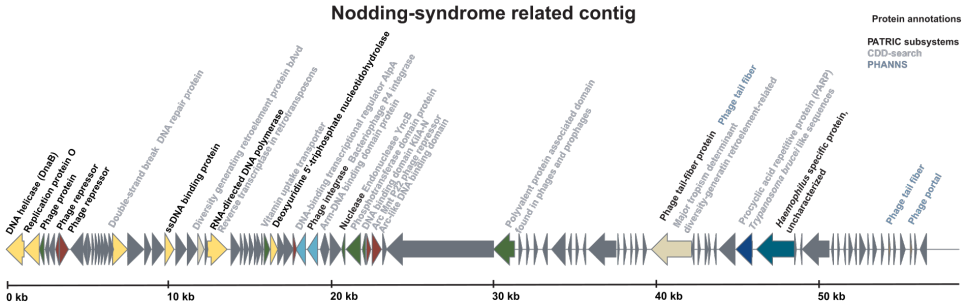


Figure 3.4. Halophage annotation

3.4 METHODS

The libraries were cross assembled [29] by concatenating all the reads into two files (forward and reverse). Low quality read pairs were removed using PRINSEQ++ (see chapter 2) [13] (`-min_len 50 -min_qual_mean 30 -entropy=0.5`). Overall, 97.8% (43,341,616 R1 reads and 44,322,504 R2 reads) passed quality control and were assembled using Metaspades [7] v3.12 (–only-assembler -K21,33,55). This resulted in 33,142 contigs larger than 800 bp. Quality of the assembly was assessed using Quast [39]. The raw reads of all libraries were

mapped against the large contigs using bowtie [58] (–sensitive -p 20 –no-unal).

Table 3.3. Cross assembly stats

# contigs ($>$ 800 bp)	33,142
# contigs ($\geq$ 1,000 bp)	24,838
# contigs ($\geq$ 5,000 bp)	3,238
# contigs ($\geq$ 10,000 bp)	1,485
# contigs ($\geq$ 25,000 bp)	533
# contigs ($\geq$ 50,000 bp)	218
Total length ($>$ 800 bp)	108,438,303
Total length ($\geq$ 1,000 bp)	101,036,476
Total length ($\geq$ 5,000 bp)	60,044,439
Total length ($\geq$ 10,000 bp)	47,990,624
Total length ($\geq$ 25,000 bp)	33,487,583
Total length ($\geq$ 50,000 bp)	22,820,351
Largest contig	671,078
Total length	108,438,303
GC (%)	63.54
N50	6,922
N75	1,980
L50	2,239
L75	10,410
# N's per 100 kbp	0

MEGAN [46] was used to annotate those contigs by parsing the results of a blastn search against the non redundant nt database (NCBI's database that includes sequences from the Nucleotide Sequence Database Collaboration and RefSeq sequences [73]) and assigning each contig to a node on the taxonomic tree. For example, if a contig only has hits against *E. coli* W3110, it will be assigned to that terminal node. On the other hand, if a contig has hits against several distinct γ-proteobacteria, it will be assigned to the internal node "gammaproteobacteria".

All original reads were mapped to the assembled contigs to construct the contig/hit table where columns represent metagenomes, rows represent contigs and entries represent

how many read pairs in that virome mapped to that contig (a read pair maps to a contig if either of the two reads maps to that contig).

Each column in the contig/hit table was RPT normalized (divide by 1,000 times the number of reads in that virome) so that the numbers in the table represent hits per 1,000 read pairs. Columns were split in two classes, cases (18 CSF, 18 buffy coat and 18 plasma) and control (9 plasma) to train a Random Forest model using the 'randomForest" [97] package in R (importance=TRUE ,ntree=2000). This model has a 9.25% out of bag error rate (3 control samples are misclassified as cases, see figure 3.5-A). To reduce the noise induced by the large number of contigs (it is unlikely that many of the 33,142 contigs are relevant to the model) the 100 contigs with the highest mean decrease accuracy measure (i.e. the 100 contigs that contributed most information to the random forest) were used to train a new Random Forest model (importance=TRUE ,ntree=300). This model has a 1.59% out of bag error rate (1 control sample is misclassified as a case, see figure 2). The top 100 contigs were sorted according to their mean decrease accuracy in the second model and annotated using MEGAN [46]. Table S1 contains the t-test p-value, MEGAN annotation, importance (mean decrease accuracy), and rank for each of the 100 top conigs. Figure S1 has a similar graph to figure 4 for the top 100 contigs.

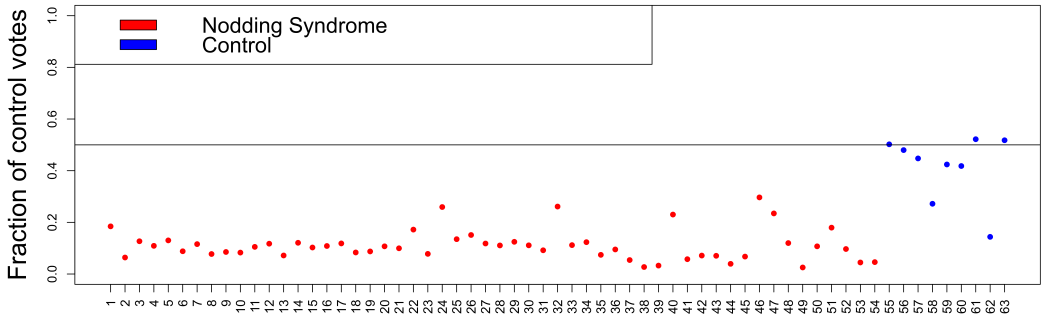
The halophage (Figure 3.4) was identified by mapping the top 100 contigs, using blast, to the assembly graph of the full cross assembly using Bandage [108]. Open Reading Frames were obtained through PATRIC [105] using the bacteriophages gene annotation recipe which uses PHANOTATE [68]. Gene annotations were obtained from PATRIC subsystems, the Conserved Domains Database search [64] and PHANNS [14]. Genome map was generated using EasyFig [92].

3.5 DISCUSSION

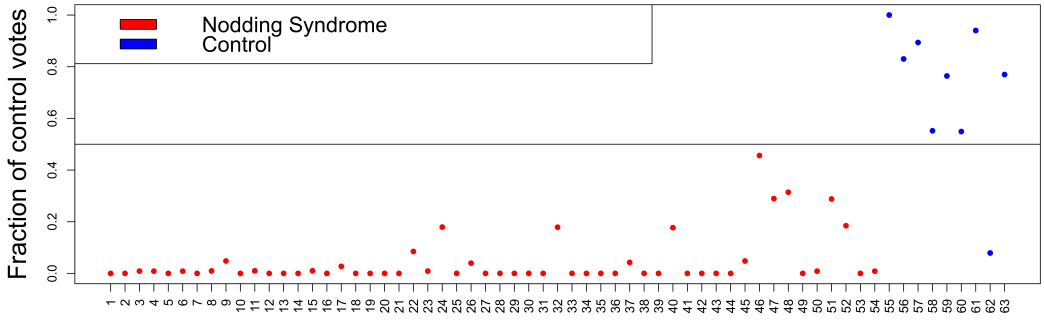
While the epidemiological association between *Onchocerca volvulus* and NS has been consistently demonstrated, many studies trying to establish a causal link have been inconclusive [28, 34]. Still, the question of why NS is not found in many areas where onchocerciasis is common remains. It has been hypothesized [18] that NS might be caused by a pathogen that also has *Simulium spp*. as a vector. It could be a filarial parasite closely related to *O. volvulus*, an alternative or additional endosymbiont to *Wolbachia*, or even a virus. With more than 1,700 *Simulium* species described, it is not unreasonable to have a blackfly species whose distribution matches NS incidence.

We were unable to identify any sequence from any *Wolbachia* species or from *Onchocerca volvulus*. We identified sequences from *Onchocerca flexuosa*. *Onchocerca flexuosa* unique as it is the only member of the *Onchocerca* genus that does not require an *Wolbachia* endosymbiont [69]. The idea that Nodding syndrome might be caused by *Onchocerca flexuosa* when it has *Halomonas* as an endosymbiont invites further exploration.

A



B



C

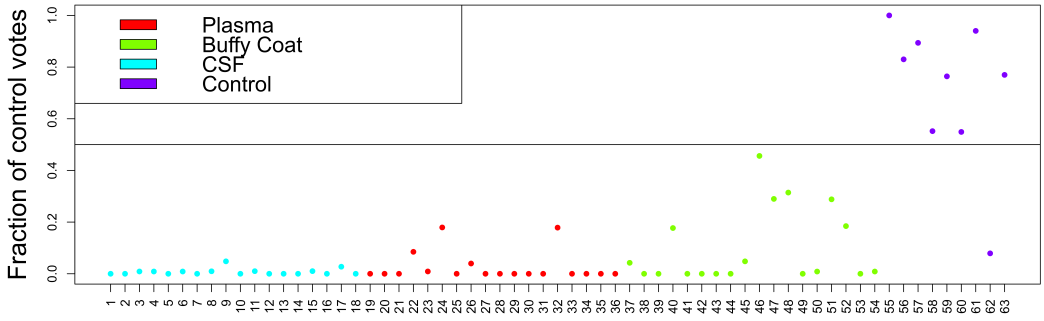


Figure 3.5. Proportion of control votes for different RF models

CHAPTER 4

PHANNS

For any given phage genome, me are unable to assign function to 50-90% [30] of genes using similarity searches. Yet, Phage have analogous structural proteins with similar 3D structure that are needed to infect and replicate in their host. In this work, we present PhANNs (PHage ANNs) an Artificial Neural Networks to classify any phage ORF into one of eleven structural classes. We use a database of 538,213 manually curated phage protein sequences and we reach f1-score of 0.87.

This Chapter consist of a brief introduction to complement the one on the paper, a copy of the peer-reviewed paper "PhANNs,a fast and accurate tool and web server to classify phage structural proteins" and a discussion expanding on the context of this tool and exploring data and experiments that were cut from the paper or send to supplementary material.

4.1 INTRODUCTION

Phages, or viruses that infect bacteria, are the most common biological entity on the Earth [40]. Yet, we are unable to assign function to 50-90% of their genes. This is mainly due to the fact that most methods to elucidate gene function are based on homology, but phages have no common origin. Nevertheless, phages across distinct groups encode analogous structural proteins that performs the same function.

Artificial Neural Networks (ANN) are proven universal approximators of functions in $\mathbb{R}^n$ [22], including the function that maps features extracted from a phage protein sequence to its structural class. In this work, we construct a well curated database of phage structural proteins and use it to train a feed-forward ANN to assign any phage protein to one of eleven

classes (ten structural plus ”others”). Furthermore, we developed a webserver where protein sequences can be uploaded for classification. The full database, as well as the code for PhANNies and the webserver, are available for download at https://edwards.sdsu.edu/phanns.

4.1.1 Phages

Phages are composed of capsid proteins that encapsulate their genome. A portal protein is used to pack the DNA or RNA genome inside the capsid. Some phages also have a complex structure called tail attached to the capsid by Collar Proteins and/or Head-Tail joining proteins. Tail itself is composed of tail proteins and/or tail shaft. Furthermore tail fibers might be attached to the tail. Figure 4.1 shows a phage.

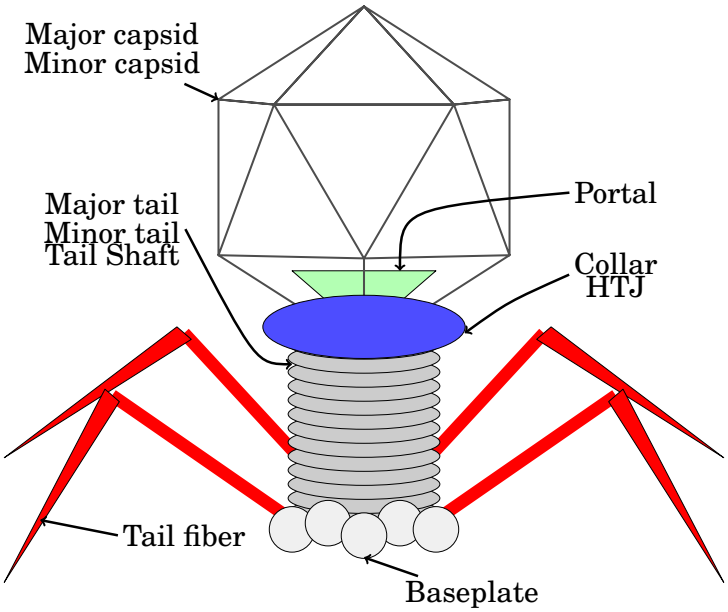


Figure 4.1. Phage structural proteins

4.1.2 Artificial neural networks

An artificial neural network is a mathematical model loosely inspired by the human brain. It consist of several neurons (aka perceptrons) linked together. Each neuron takes inputs either form the model of from other neurons, performs some operation on them (traditionally a lineal combination) and returns some output.

The multi-layer ANN architecture consists of ordered groups (aka layers) of neurons that are all connected to every neuron on the previous and next group. No connections are made within a layer. The first layer is the input layer and the last one is the output layer (see Figure 4.2)

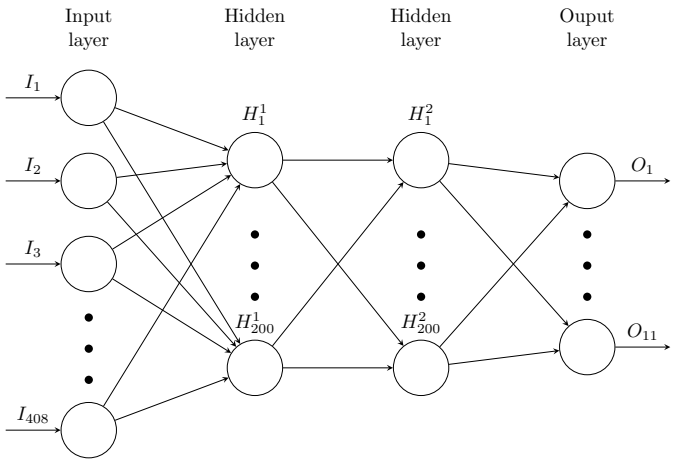


Figure 4.2. A multi-layer ANN

The process of ”training” an ANN consist of tuning the models parameters (the coefficients of the lineal combination in this case) to make the input and known output to match closely for a large set of known training examples.

4.2 PAPER

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

PhANNs, a fast and accurate tool and web server to classify phage structural proteins

Vito Adrian Cantu[1,2], Peter Salamon[2,3], Victor Seguritan[1¤a], Jackson Redfield[4¤b], David Salamon[3], Robert A. Edwards[1,2,4¤c], Anca M. Segall[1,2,4]*

1 Computational Science Research Center, San Diego State University, San Diego, United States of America, **2** Viral Information Institute, San Diego State University, San Diego, United States of America, **3** Department of Mathematics and Statistics, San Diego State University, San Diego, United States of America, **4** Department of Biology, San Diego State University, San Diego, United States of America

¤a Current address: Experian, Costa Mesa, CA, United States of America
¤b Current address: Inova Diagnostics, San Diego, CA, United States of America
¤c Current address: College of Science and Engineering, Flinders University, South Australia
* asegall@sdsu.edu

OPEN ACCESS

Citation: Cantu VA, Salamon P, Seguritan V, Redfield J, Salamon D, Edwards RA, et al. (2020) PhANNs, a fast and accurate tool and web server to classify phage structural proteins. PLoS Comput Biol 16(11): e1007845. https://doi.org/10.1371/journal.pcbi.1007845

Editor: Mihaela Pertea, Johns Hopkins University, UNITED STATES

Received: March 30, 2020

Accepted: September 26, 2020

Published: November 2, 2020

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1007845

Data Availability Statement: The full database, as well as the code for PhANN and the webserver, are available for download at http://edwards.sdsu.edu/

Abstract

For any given bacteriophage genome or phage-derived sequences in metagenomic data sets, we are unable to assign a function to 50–90% of genes, or more. Structural protein-encoding genes constitute a large fraction of the average phage genome and are among the most divergent and difficult-to-identify genes using homology-based methods. To understand the functions encoded by phages, their contributions to their environments, and to help gauge their utility as potential phage therapy agents, we have developed a new approach to classify phage ORFs into ten major classes of structural proteins or into an "other" category. The resulting tool is named PhANNs (Phage Artificial Neural Networks). We built a database of 538,213 manually curated phage protein sequences that we split into eleven subsets (10 for cross-validation, one for testing) using a novel clustering method that ensures there are no homologous proteins between sets yet maintains the maximum sequence diversity for training. An Artificial Neural Network ensemble trained on features extracted from those sets reached a test F_1-score of 0.875 and test accuracy of 86.2%. PhANNs can rapidly classify proteins into one of the ten structural classes or, if not predicted to fall in one of the ten classes, as "other," providing a new approach for functional annotation of phage proteins. PhANNs is open source and can be run from our web server or installed locally.

Author summary

Bacteriophages (phages, viruses that infect bacteria) are the most abundant biological entity on Earth. They outnumber bacteria by a factor of ten. As phages are very different from each other and from bacteria, and we have relatively few phage genes in our database compared to bacterial genes, we are unable to assign function to 50–90% of phage genes. In this work, we developed PhANNs, a machine learning tool that can classify a phage

phanns and https://github.com/Adrian-Cantu/PhANNs.

Funding: This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Army Research Office (ARO) under cooperative Agreement Number W911NF-17-2-0105, and awarded as a partial subcontract to AMS. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, ARO, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This work was supported by NIH grant RC2DK116713 to AMS and RAE and US Department of Defense: Defense Threat Reduction Agency grant number DTRA13081-32220 to RAE. Victor Seguritan and Jackson Redfield were supported by NSF DMS 0827278 Undergraduate BioMath Education grant awarded to AMS and PS. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

gene as one of ten structural roles, or "other". This approach does not require a similar gene to be known.

This is a *PLOS Computational Biology* Software paper.

Introduction

Bacteriophages (phages) are the most abundant biological entity on the Earth [1]. They modulate microbial communities in several possible ways: by lysing specific taxonomic members or narrow groups of microbiomes, they affect the microbial population dynamics and change niche availability for different community members. Via transduction and/or lysogeny, they mediate horizontal transfer of genetic material such as virulence factors [2], metabolic auxiliary genes [3], photosystems and other genes to enhance photosynthesis[4], and phage production in general, by providing the host with immunity from killing by other phages. Temperate phages can become part of the host genome as prophages; most bacterial genomes contain at least one, and often multiple prophages [5,6].

Phage structures (virions) are composed of proteins that encapsulate and protect their genomes. The structural proteins (or virion proteins) also recognize the host, bind to its surface receptors and deliver the phage's genome into the host's cell. Phage proteins, especially structural ones, vary widely between phages and phage groups, so much so that sequence alignment based methods to assign gene function fail frequently: we are currently unable to assign function to 50–90% of phage genes [7]. Experimental methods such as protein sequencing, mass spectrometry, electron microscopy, or crystallography, in conjunction with antibodies against individual proteins, can be used to identify structural proteins but are expensive and time-consuming. A fast and easy-to-use computational approach to predict and classify phage structural proteins would be highly advantageous as part of pipelines for identifying functional roles of proteins of bacteriophage origins. The current increased interest in using phages as therapeutic agents [8,9] motivates annotations for as much of the phage genome as possible. Even if they are somewhat tentative and not experimentally validated, annotations of the relatively non-toxic structural proteins versus the potentially host health-threatening toxins and other virulence factors could inform decisions whether to choose one specific phage versus another.

Machine learning has been used to attack similar biological problems. In 2012, Seguritan et al. [10] developed Artificial Neural Networks (ANNs) that used normalized amino acid frequencies and the theoretical isoelectric point to classify viral proteins as structural or not structural with 85.6% accuracy. These ANNs were trained with proteins of viruses from all three domains of life. They also trained two distinct ANNs to classify phage capsid versus phage non-capsid ORFs and phage "tail associated" versus phage "non-tail-associated" ORFs. Subsequently, several groups have used different machine learning approaches to improve the accuracy of predictions. The resulting tools are summarized in **Table 1.**

Each of these previous approaches has important limitations: 1) The classification is limited to two classes of proteins (e.g.,"capsid" or "not capsid"). 2) Training and testing sets were small (only a few hundred proteins in some cases), limiting the utility of these approaches beyond those proteins used in testing. 3) Methods that rely on predicting secondary structure (e.g., VIRALpro [11]) are slow to run. In general, these newer methods have improved accuracy at the cost of lengthening the time required for training, or have used very small training and/or test sets.

Table 1. Summary of previous ML-based methods for classifying viral structural proteins.

Reference	Method	Target proteins	Database size	Accuracy
Seguritan et al.[10]	ANN	structural (all viruses) versus non-structural (all viruses)	6,303 structural	85.6%
			7,500 non-structural	
Seguritan et al.[10]	ANN	capsid versus non-capsid (phages only)	757 capsid	91.3%
			10,929 non-capsid	
Seguritan et al.[10]	ANN	Tail-associated versus non-tail (phages only)	2,174 tail	79.9%
			16,881 non-tail	
Feng et al.[33]	Naïve Bayes	structural versus non-structural	99 structural	79.15%
			208 non-structural	
Zhang et al.[34]	Ensemble Random Forest	structural versus non-structural	253 structural	85.0%
			248 non-structural	
Galiez et al.[11]	SVM	capsid versus non-capsid	3,888 capsid	96.8%
			4,071 non-capsid	
Galiez et al.[11]	SVM	tail versus non-tail	2,574 tail	89.4%
			4,095 non-tail	
Manavalan et al.[35]	SVM	structural versus non-structural	129 structural	87.0%
			272 non-structural	
This work	ANN	Ten distinct phage structural classes plus "others"	168,660 structural	86.2%
			369,553 non-structural	

https://doi.org/10.1371/journal.pcbi.1007845.t001

Artificial Neural Networks (ANN) are proven universal approximators of functions in $\mathbb{R}^n$ [12], including the mathematical function that maps features extracted from a phage protein sequence to its structural class. We have constructed a manually-curated database of phage structural proteins and have used it to train a feed-forward ANN to assign any phage protein to one of eleven classes (ten structural classes plus a catch-all class labeled "others"). Furthermore, we developed a web server where protein sequences can be uploaded for classification. The full database, as well as the code for PhANNs and the webserver, are available for download at http://edwards.sdsu.edu/phanns and https://github.com/Adrian-Cantu/PhANNs

Methods

Database

In this work, we generated two complementary protein databases, "classes" and "others". The "classes" database contains curated sequences of ten phage structural functions (Major capsid, Minor capsid, Baseplate, Major tail, Minor tail, Portal, Tail fiber, Tail sheath, Collar, and Head-Tail Joining). These functional classes are not exhaustive (and we will add more classes in the future); they represent the dominant structural protein roles present in most (but not all) phages [13]. The terms/descriptors for these classes are addressed in the next section. Major capsid proteins are those that form the phage head. Many but not all phages also encode minor capsid proteins that decorate and/or stabilize the head or proteins present at the vertices of the icosahedral heador at the center of the hexon faces. Portals form a ring at the base of the phage head and serve to dock the packaging complex that translocates the genome into the phage head. Head-tail joining (aka head-tail connector or head completion) proteins form rings inserted between the portal ring and the tail. The collar is present in some phages, *e.g.* the Lactococcal phages, at the base of the neck/top of the tail to which the so-called whiskers attach. Major tail proteins form the inner tail tube of the tailed phages, whereas the tail sheath (aka the tail shaft) proteins form the outside of the tail, and permit contraction. Minor tail

proteins may comprise several kinds of proteins associated with the tail, including the tape measure protein. Baseplate proteins are those that are attached to the tail and to which the tail fibers are attached, the latter being a relatively common determinant of host range. The "others" database contains all phage ORFs that do not encode proteins annotated as "structural" or as any of the ten categories above.

The database of "classes"

Sequences from the ten structural classes were downloaded from NCBI's protein database using a custom search for the class title (the queries are in the "ncbi_get_structural.py" script in the GitHub repository). Curation consisted of grouping sequences by their description (part of the fasta header) and deciding which descriptions to include. The list of included headers for each class can be found here https://github.com/Adrian-Cantu/PhANNs/tree/master/model_training/01_fasta; the variations of terms included are too many to be included here. All the terms preceded by a "+" (or "+ +") were included in the respective database. In the particular case of tail fibers, we did not include the descriptions "phage tail fiber assembly protein" (3,662 proteins) nor many "partial protein" variations (1,500+ proteins).

This method for collecting data has the limitation that a proportion of phage sequences in the database are misannotated and that NCBI has no controlled vocabulary for bacteriophage protein functions so it is occasionally difficult to account for misspelled annotations and/or alternative naming. However, it is clear from previous machine learning applications that a larger number of training examples is more important for optimal model performance than a perfectly curated training set [14]. To minimize inclusion of wrongly annotated protein sequences, we manually curated the databases to address these limitations.

The "others" database

To generate a database for the "others" class, all available phage genomes (8,238) were downloaded from GenBank on 4/13/19. ORFs were found using the GenBank PATRIC [15] server with the phage recipe [16]. Sequences annotated as structural or any of the ten classes were removed during manual curation. Furthermore, the remaining sequences were de-replicated at 60% together with sequences in the "classes" database using CD-hit [17]. Any phage ORF that clustered with a sequence from the "classes" database was removed from the "others" database.

Training, test, and validation split

Sequences in each class were clustered at 40% using CD-hit and split into eleven sets (10 for cross validation and one for testing, as shown in **Fig 1**). Once the clusters were established, to prevent loss of the sequence diversity available within the clusters, which is essential for optimal training, the clusters were expanded by adding back *within* each set all the representatives of that set (described in **Fig 1**). Subsequently, the sets corresponding to each structural class were merged. We named the generated sets 1D-10D and TEST. Splitting the database this way ensures that the different sets share no homologous proteins while recapturing all the sequence diversity present in each class. Finally, 100% dereplication was performed to remove identical sequences (See **Table 2**). The effect of the cluster expansion on performance is explored in **S1 and S2 Figs**.

Extraction of features

The frequency of each dipeptide (400 features) and tripeptide (8,000 features) was computed for each ORF sequence in both the "classes" and "others" databases. As a potential time-saving

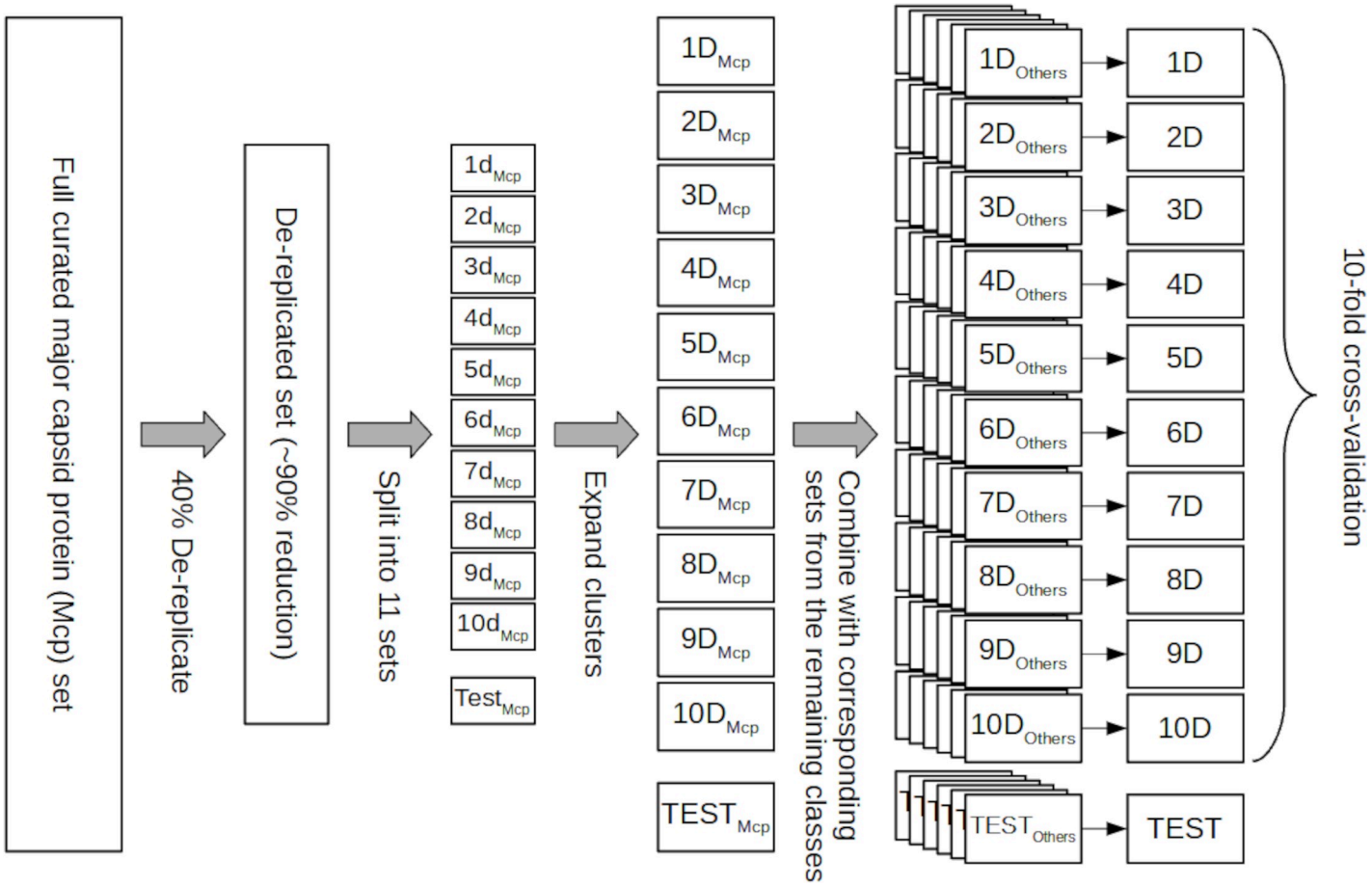


Fig 1. Non homologous database split—To ensure that no homologous sequences are shared between the test, validation, and training sets the sequences from each class (Major capsid proteins in this figure) were de-replicated at 40%. In the de-replicated set, no two proteins have more than 40% identity and each sequence is a representative of a larger cluster of related proteins. The de-replicated set is then randomly partitioned into eleven equal size subsets, ($1d_{Mcp}$-$10d_{Mpc}$ plus $Test_{Mpc}$). Those subsets are expanded by replacing each sequence with all the sequences in the cluster it represents (subsets $1D_{Mpc}$-$10D_{Mpc}$ plus $TEST_{Mpc}$). Analogous subsets are generated for the remaining ten classes and corresponding subsets are combined to generate the subsets used for 10-fold cross-validation and testing (1D-10D and TEST).

https://doi.org/10.1371/journal.pcbi.1007845.g001

Table 2. Database numbers—Raw sequences were downloaded using a custom script available at https://github.com/Adrian-Cantu/PhANNs. All datasets can be downloaded from the web server. *Numbers before and after removing sequences at least 60% identical to a protein in the classes database.

Class	Raw sequences	After manual curation	After de-replication at 40%	After expansion and de-replication at 100%
Major capsid	112,987	105,653	1,945	35,755
Minor capsid	2,901	1,903	261	1,055
Baseplate	75,599	19,293	401	6,221
Major tail	66,513	35,030	536	7,704
Minor tail	94,628	80,467	918	18,002
Portal	210,064	189,143	2,310	59,745
Tail fiber	29,132	18,514	1,222	7,256
Tail sheath	37,885	35,570	599	15,349
Collar	4,224	3,709	339	2,105
Head-Tail joining	60,270	58,658	1,317	15,468
Total structural	**694,203**	**547,940**	**9,848**	**168,660**
Others	733,006	643,735/643,380*	106,004	369,553

https://doi.org/10.1371/journal.pcbi.1007845.t002

procedure during neural net training while also permitting classification of more diverse sequences, each amino acid was assigned to one of seven distinct "side chain" chemical groups (S1 Table). The frequency of the "side chain" 2-mers (49 features), 3-mers (343 features), and 4-mers (2,401 features) was also computed. Finally, some extra features, namely isoelectric point, instability index (whether a protein is likely to degrade rapidly; [18]), ORF length, aromaticity (relative frequency of aromatic amino acids; [19]), molar extinction coefficient (how much light the protein absorbs) using two methods (assuming reduced cysteins or disulfide bonds), hydrophobicity, GRAVY index (average hydropathy; [20]) and molecular weight, were computed using Biopython [21]. All 11,201 features were extracted from each of 538,213 protein sequences. The complete training data set can be downloaded from the web server (https://edwards.sdsu.edu/phanns).

ANN architecture and training

We used Keras [22] with the TensorFlow [23] back-end to train eleven distinct ANN models using a different subset of features. We named the models to indicate which feature sets were used in training: the composition of 2-mers/dipeptides (di), 3-mers /tripeptides (tri) or 4-mer/tetrapeptide (tetra), or side chain groups (sc) (as shown in S1 Table), and whether we included the extra features (p) or not. A twelfth ANN model was trained using all the features (Table 3).

Each ANN consists of an input layer, two hidden layers of 200 neurons, and an output layer with 11 neurons (one per class). A dropout function with 0.2 probability was inserted between layers to prevent overfitting. ReLU activation (to introduce non-linearity) was used for all layers except the output, where softmax was used. Loss was computed by categorical cross-entropy and the ANN is trained using the "opt" optimizer until 10 epochs see no training loss reduction. The model at the epoch with the lowest validation loss is used. Class weights inversely proportional to the number of sequences in that class were used.

10-fold cross-validation. Sets 1D to 10D (see Fig 1) were used to perform 10-fold cross-validation; ten ANNs were trained as described above, sequentially using one set as the validation set and the remaining nine as the training set. The results are summarized in **Figs 2, 3, 4, S1 and S2.**

Table 3. Feature types included in each of the 12 models. di—2-mer/dipeptide composition; **tri**—3-mer/tripeptide composition; **tetra**—4-mer/tetrapeptide composition; sc—side-chain grouping; **p**—plus all the extra features [isoelectric point, instability index (whether a protein is likely to be degraded rapidly), ORF length, aromaticity (relative frequency of aromatic amino acids), molar extinction coefficient (how much light a protein absorbs) using two methods (assuming reduced cysteines or disulfide bonds), hydrophobicity, GRAVY index (average hydropathy), and molecular weight, as computed using Biopython. - *Per class score figures are available as supplementary material.

Model	di	tri	di_sc	tri_sc	tetra_sc	p
di_sc*			x			
di_sc_p*			x			x
tri_sc*				x		
tri_sc_p*				x		x
tetra_sc*					x	
tetra_sc_p*					x	x
di	x					
di_p	x					x
tri		x				
tri_p		x				x
tetra_sc_tri_p		x			x	x
all	x	x	x	x	x	x

https://doi.org/10.1371/journal.pcbi.1007845.t003

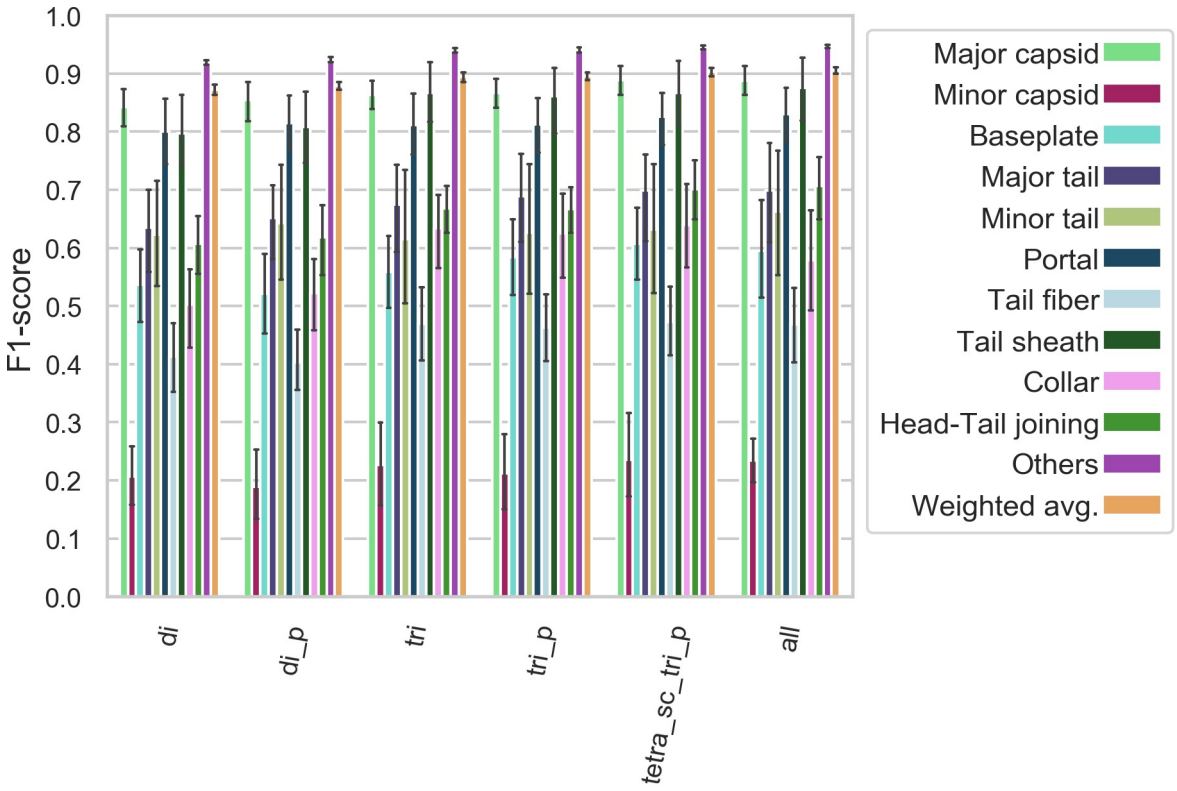


Fig 2. Model-specific F_1 score—F_1 scores (harmonic mean of precision and recall) for each polypeptide model/class combination. All models follow similar trends as to which classes are more or less difficult to classify correctly. Error bars represent the 95% confidence intervals.

https://doi.org/10.1371/journal.pcbi.1007845.g002

The PhANNs score. For each input sequence, PhANNs run 10 ANNs predictions (those trained during the 10-fold cross-validation). Each of those 10 ANNs outputs the soft-max scores for every class (a number between 0 and 1, such that the score of all classes adds to 1). PhANNs outputs the per class sum of the ten ANNs scores (the maximum achievable PhANNs score is 10, as there are ten ANNs). The input sequence is classified as the class with the highest PhANNs score.

To give a clearer indication of the quality of this prediction we added a "confidence" score to each prediction. The "confidence" score shows what fraction of sequences in the test set that were classified as the same class as the input sequence, and with the same PhANNs score or higher, were correctly classified (True positives). The confidence scores differ depending on the protein class. For example, a sequence classified as "major capsid" with a PhANNs score of 7 has 97% confidence, while a "Tail fiber" with a PhANNs score of 7 has only 82.4% confidence. The per class relationship between the PhANNs score and the confidence is explored in Fig 5.

Web server

We developed an easy-to-use web server for users to upload and classify their own sequences. Although ANNs need substantial computational resources for training the model (between 54,861 and 127,756,413 parameters need to be tuned, depending on the model), the trained model can make fast *de novo* predictions. Our web server (https://edwards.sdsu.edu/phanns) can predict the structural class of an arbitrary protein sequence in seconds and assign all the

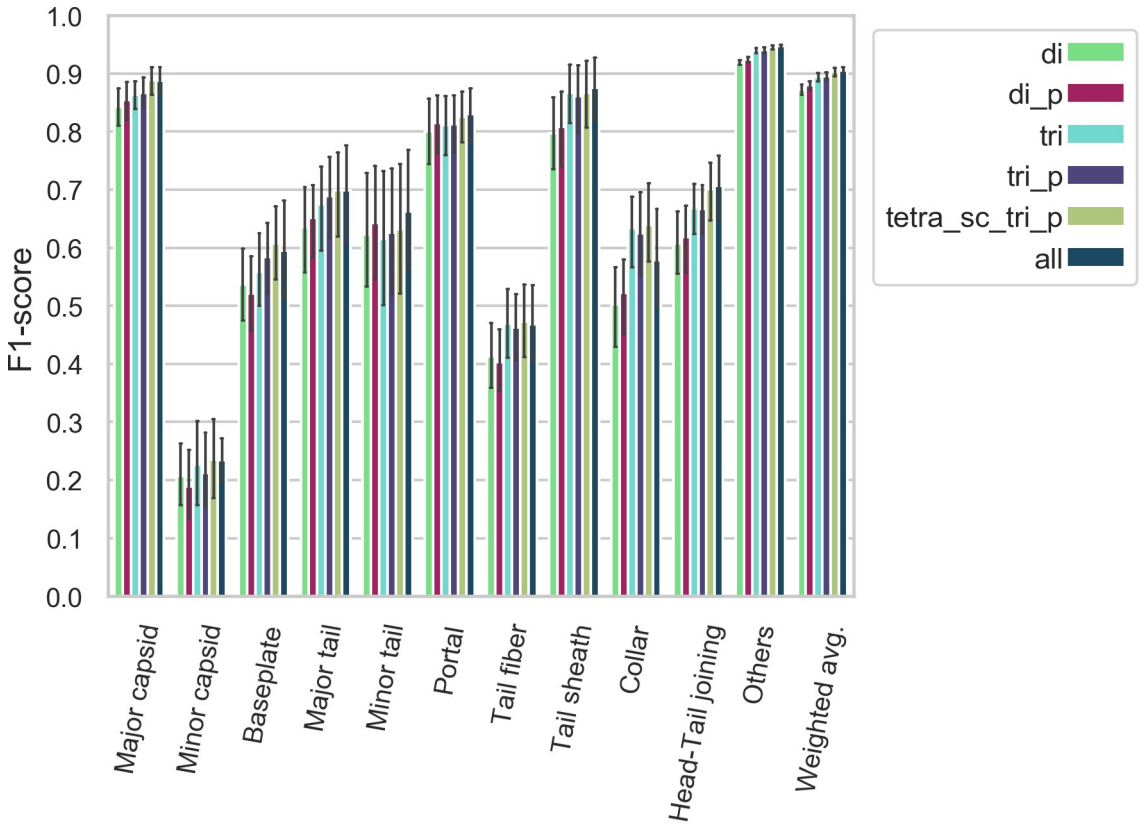


Fig 3. Class-specific F_1 score—F_1 scores (harmonic mean of precision and recall) for each polypeptide model/class combination. Some classes, such as minor capsid, tail fiber, or minor tail, are harder to classify correctly irrespective of the model used. Error bars represent the 95% confidence intervals.

https://doi.org/10.1371/journal.pcbi.1007845.g003

ORFs in a phage genome to one of the 10 classes in minutes. The application can also be downloaded and run locally for large numbers of queries or if privacy is a concern.

Results and discussion

We evaluated the performance of 120 ANNs (10 per model type) on their respective validation set. For each ANN, we computed the precision, recall, and F_1-score of the 11 classes. A "weighted average" precision, recall and F_1-score, where the score for each class is weighted by the number of proteins in that class (larger classes contribute more to the score) was computed. The accuracy (fraction of observation correctly classified) is equivalent to the weighted average recall. The three weighted average scores are represented as a 12th class. This gives us ten observations for each combination of model type and class, which allows us to construct the confidence intervals depicted in **Figs 2, 3 and 4.**

(Figs 2 and S1) shows that all the models follow the same trend as to which classes they predict with higher or lower accuracy. Some classes of proteins, for example major capsids, collars, and head-tail joining proteins, are predicted with high accuracy. On the other hand, the minor capsid and tail fiber protein classes seem to be intrinsically hard to predict with high accuracy irrespective of the model type used (**Figs 3 and S2**). One reason for this is the limited size of the training set: the minor capsid protein set is the smallest class, with only 581 proteins available for inclusion in our database. Even if the classes were weighted according to their size during training, it appears we do not have enough training examples to identify this class with

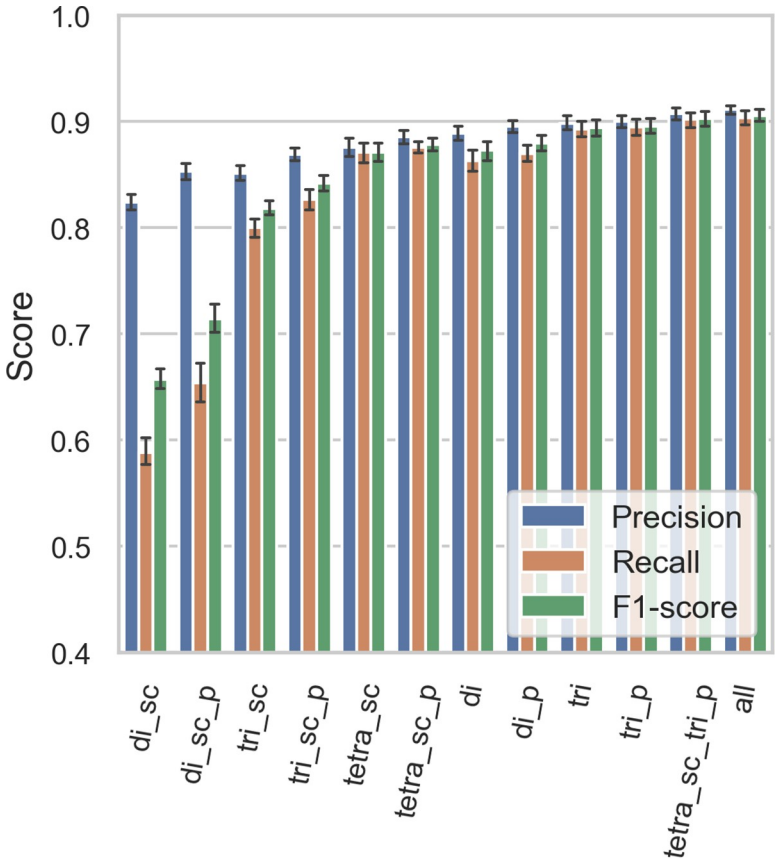


Fig 4. Model-specific validation weighted average scores—Precision, recall, and F_1 scores for all models. Precision is higher in all models as the "others" class is the largest and easiest to classify correctly. Error bars represent the 95% confidence intervals.

https://doi.org/10.1371/journal.pcbi.1007845.g004

high accuracy. Furthermore, "minor capsid" is often misclassified as "portal" (Fig 6). This probably reflects an annotation bias, as we found about 800 proteins annotated as "portal (minor capsid)" in the raw sequences. When the ~800 proteins are analyzed with PhANNs, over 90% are predicted to be portal proteins. Although these were removed during manual curation of the training data sets, some (small) fraction of minor capsid proteins in our database may have been annotated as "minor capsid" by homology to one of those 800 sequences.

The predictive accuracy for a specific class of proteins is likely to be affected by the bias in the training datasets. The bias could be biological and/or due to a sampling bias. An example of the former is the tail fiber class: the tail fiber is one of the determinants of the host range of the virus, and is under strong evolutionary selective pressure [24–29]. On the other hand, sampling bias may be introduced due to oversampling of certain types of phages, such as the thousands of mycobacterial phages isolated as part of the SEA-PHAGES project [30], many of which are highly related to each other.

Average validation F_1-scores range from 0.653 for the "di_sc" model to 0.841 for the "tetra_sc_tri" model (Fig 4). Although the average validation F_1-score for the top three models "tri_p" (0.832), "tetra_sc_tri_p" (0.841), and "all" (0.827) are not significantly different from each other, we decided to use "tetra_sc_tri_p" for the web server and all subsequent analyses because, while it uses ~7% fewer features than "all" (10,409 vs 11,201), we expect that the tetra side chain features may be better than the tripeptide features at generalizing predictions and accessing greater sequence diversity.

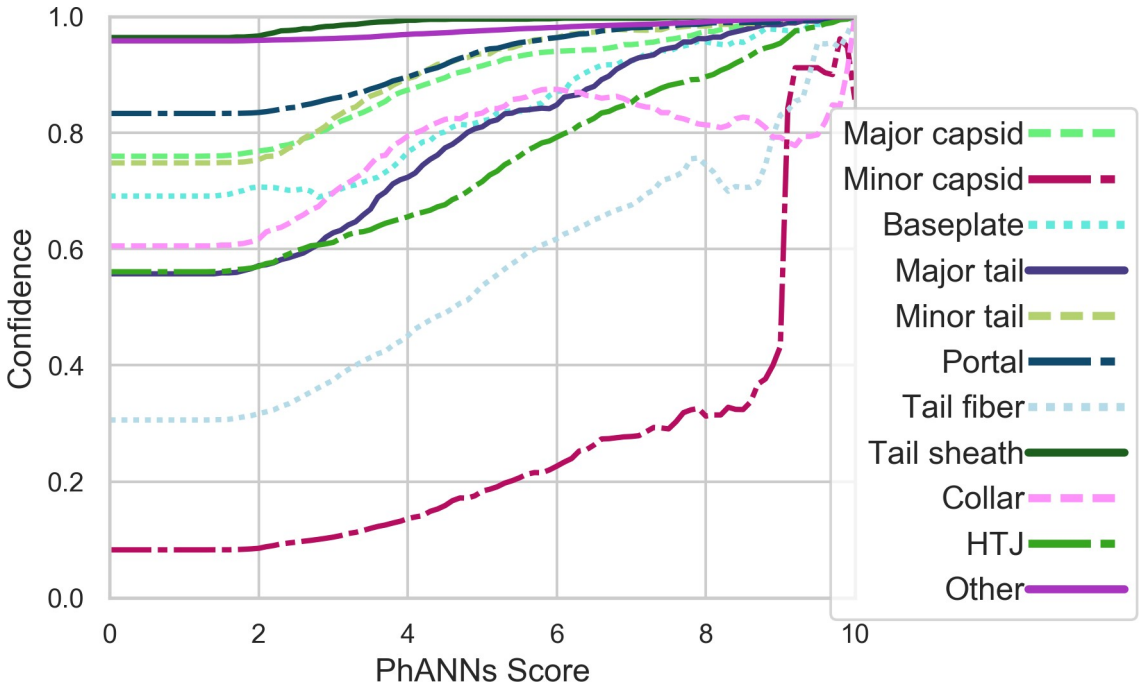


Fig 5. Per class relationship between PhANNs score and confidence—The confidence corresponding to a particular class PhANNs score represents the fraction of true positives (correctly classified) sequences in the test set that were classified as that class, with a given PhANNs score or higher. As it is uncommon for the highest class PhANNs score to be less than 2, the left side of the graph includes all test proteins that were classified as that class, and the confidence corresponds to the per class precision (see Table 4).

https://doi.org/10.1371/journal.pcbi.1007845.g005

Using the "tetra_sc_tri_p" ensemble, we predicted the class of each sequence in the test set (46,801) by averaging the scores of each of the ten ANNs. Results are summarized in **Fig 6 and Table 4.** Doing this we reach a test F_1-score of 0.89 and accuracy of 86.2% over the eleven classes.

Higher accuracy can be reached if one is willing to disregard sequences with low PhANNs scores. Using only sequences with a PhANNs score of 5 or higher, the F_1-score for the test set is 0.945, accuracy is 94%, with 9,006 of 46,801 (~20%) test sequences being "not classified". If using sequences with a PhANNs score of 8 or higher, the F_1-score for the test set is 0.982, accuracy is 98%, but 19,208 of 46,801 (~41%) test sequences would be "not classified" (see Fig 7). Table 4 shows summary statistics for the complete test set, while Table 5 shows the same statistics for the test subset of sequences with PhANNs 8 or greater. The stringency with which users interpret the PhANNs score may vary depending on their specific need. Therefore we recommend that the actual PhANNs score (or the confidence score) be reported in addition to the predicted function class.

Because "minor capsid" is the worst performing class in our test set, we trained an analogous ANN ensemble without that class to explore if accuracy of the remaining classes is improved. Multiple metrics can be used to assess which model is better. The per class ROC curves of both models [Fig 8A **(with minor capsid class) and 8-B (without minor capsid class**)] and areas under the curves are similar. Removing the minor capsid class from the models doesn't significantly alter the relationship between the PhANNs score and the confidence score (Fig 8C and 8D). The confusion matrices of both models (Fig 8E and 8F) show that predictions for portal proteins improve, as 3% of them are misclassified as minor capsid. For all other classes, the two models are similar with respect to which classes are most commonly

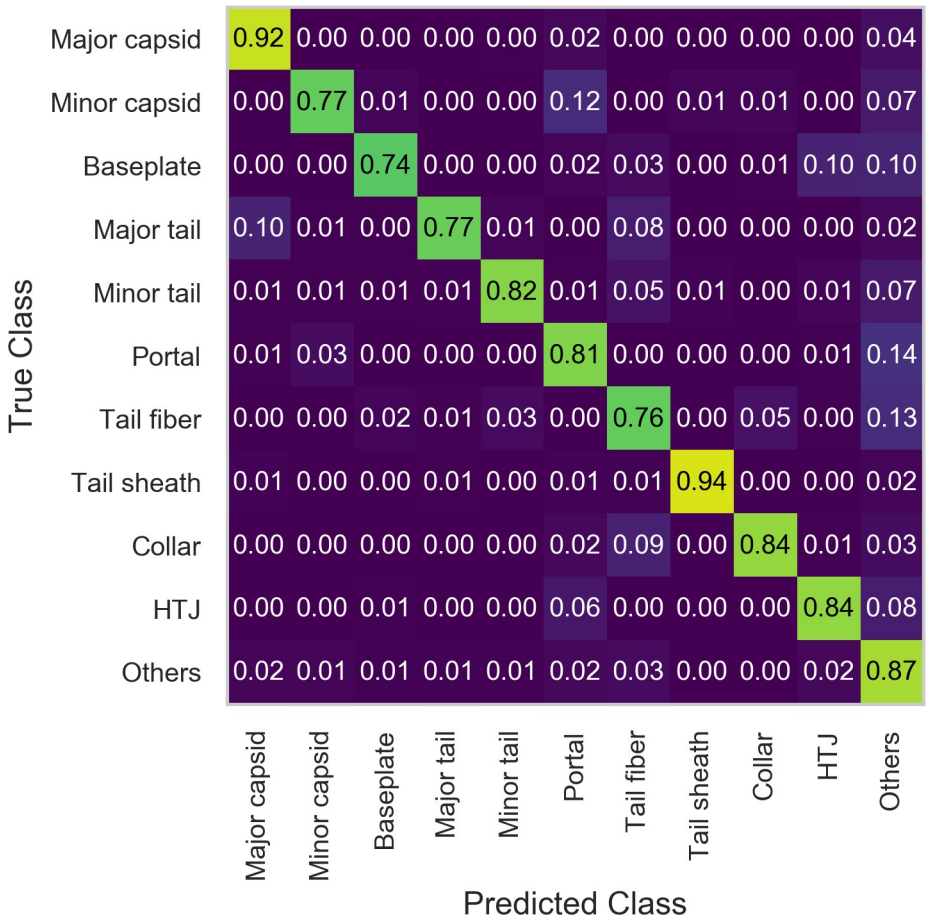


Fig 6. Confusion matrix using the "tetra_sc_tri_p" model—Each row shows the proportional classification of test sequences from a particular class. A perfect classifier would have 1 on the diagonal and 0 elsewhere. In general, a protein that is misclassified is predicted as "others".

https://doi.org/10.1371/journal.pcbi.1007845.g006

Table 4. Results of per class classification for the test set. Support indicates the number of test sequences in each specific class. accuracy (fraction of observation correctly classified) is equivalent to the weighted average recall (weighted by the support of each class). The macro average is unweighted (all classes contribute the same).

	precision	recall	f1-score	support
Major capsid	0.80	0.91	0.85	2,456
Minor capsid	0.07	0.78	0.13	81
Baseplate	0.69	0.75	0.72	851
Major tail	0.55	0.79	0.65	502
Minor tail	0.66	0.82	0.73	1,072
Portal	0.81	0.81	0.81	5,261
Tail fiber	0.35	0.74	0.47	648
Tail sheath	0.97	0.93	0.95	2,031
Collar	0.51	0.86	0.64	300
Head-Tail joining	0.56	0.84	0.67	1,277
Others	0.96	0.86	0.91	32,322
macro avg	0.63	0.83	0.68	46,801
weighted avg	0.89	0.86 (accuracy)	0.87	46,801

https://doi.org/10.1371/journal.pcbi.1007845.t004

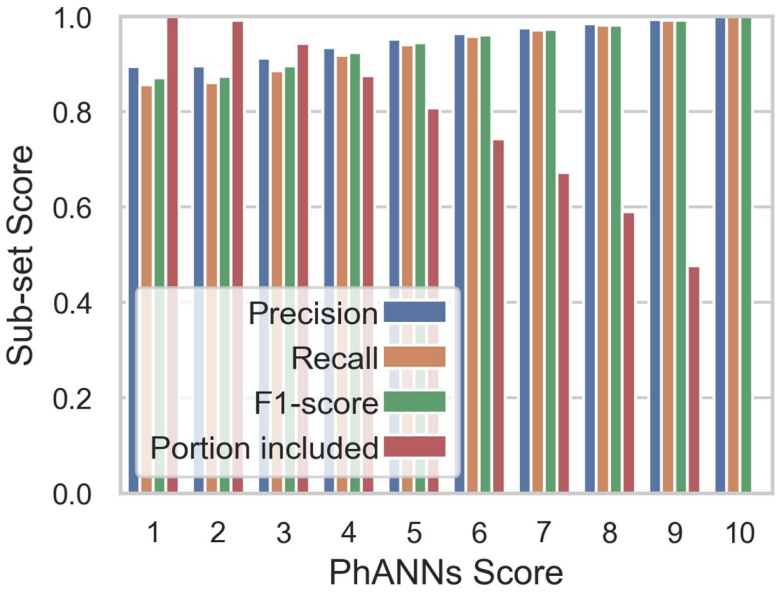


Fig 7. Effect of disregarding low scoring test proteins—Progression of the weighted average precision, recall and F_1-score of the test set after excluding low scoring proteins. The portion of included proteins is the fraction that can be classified if you only trust that score or higher. Very few test proteins have PhANNs score of 10 and not all classes are represented.

https://doi.org/10.1371/journal.pcbi.1007845.g007

confused. A comparison of per class precision, recall and F_1-score can be found in Table 6. When the minor capsid class is excluded, metrics are just as likely to improve as to worsen, and the accuracy gain is only 1%; greater accuracy gains can be achieved by disregarding sequences with low PhANNs scores as "not classified," as described above. Therefore, we decided not to exclude the minor capsid class from our model; the performance in this class is likely to improve in the future, as more sequences become available and, hopefully, are experimentally validated.

Table 5. Results of per class classification for proteins in the test set with a PhANNs score of 8 or higher. Support indicates the number of test sequences in each specific class. accuracy (fraction of observation correctly classified) is equivalent to the weighted average recall (weighted by the support of each class). The macro average is unweighted (all classes contribute the same).

	precision	recall	F_1-score	support
Major capsid	0.99	0.99	0.99	1,563
Minor capsid	0.28	0.96	0.43	45
Baseplate	0.97	0.83	0.89	151
Major tail	0.95	0.97	0.96	307
Minor tail	0.95	0.99	0.97	625
Portal	0.99	0.94	0.97	3,810
Tail fiber	0.89	0.94	0.91	360
Tail sheath	1.00	1.00	1.00	1,495
Collar	0.82	1.00	0.90	98
Head-Tail joining	0.91	1.00	0.95	916
Others	0.99	0.99	0.99	18,223
macro avg	0.89	0.96	0.91	27,593
weighted avg	0.98	0.98 (accuracy)	0.98	27,593

https://doi.org/10.1371/journal.pcbi.1007845.t005

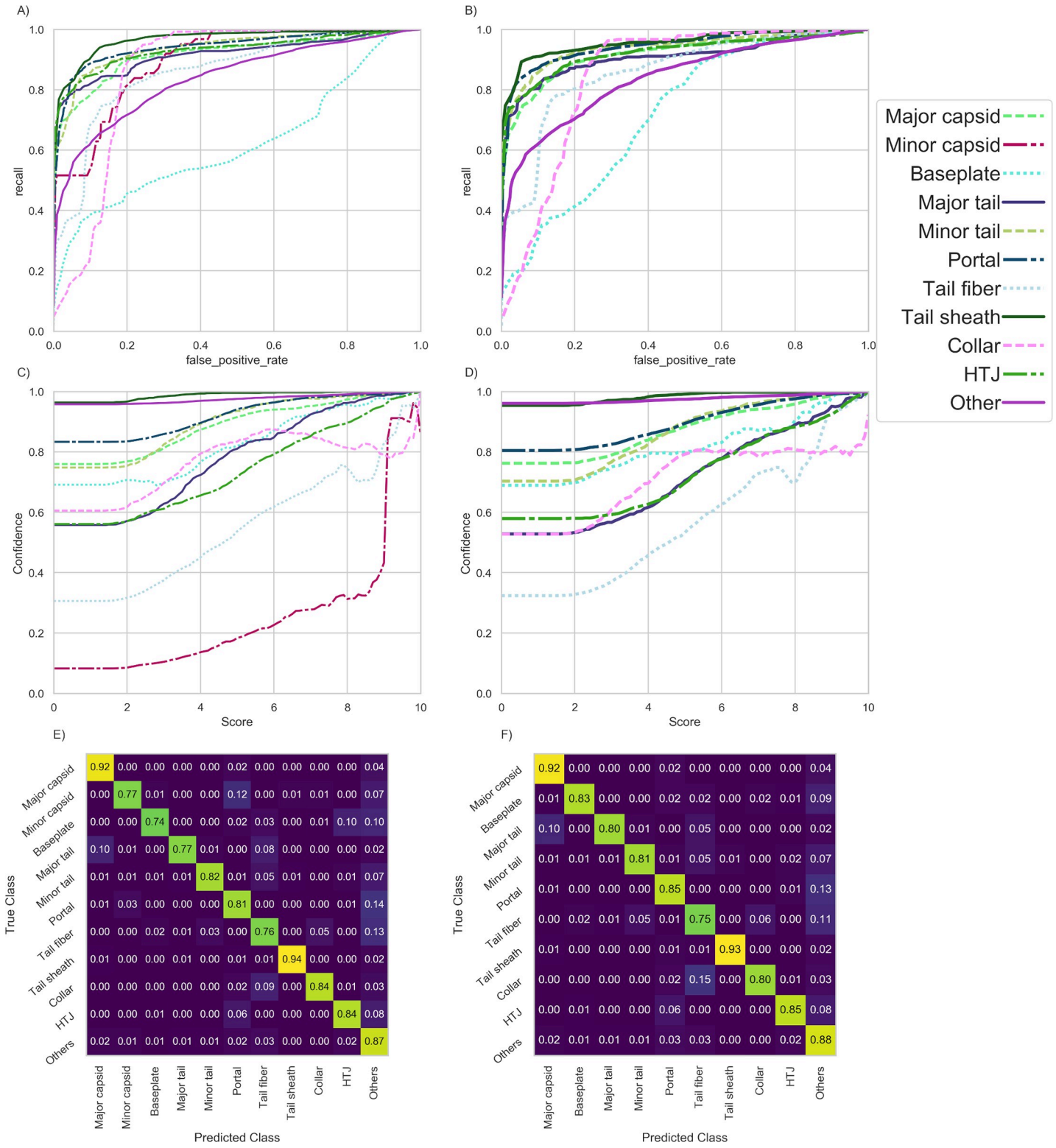


Fig 8. Comparison of "tetra_sc_tri_p" model trained with and without the Minor capsid class—As minor capsid is the worst performing class in our test set, we trained an analogous ANN ensemble with it removed. Panels A and B show the ROC curves for the models with and without minor capsid respectively. Panels C and D show the relationship between PhANNs score and Confidence for the models with and without minor capsid respectively. Panels E and F show the confusion matrix for the models with and without minor capsid respectively.

https://doi.org/10.1371/journal.pcbi.1007845.g008

Table 6. The effect on the models's scores from excluding the minor capsid class (mc)—Most scores are affected only slightly and are as likely to improve as to worsen.

	precision	precision (mc)	recall	recall (mc)	F_1-score	F_1-score (mc)	support	ROC area	ROC area (mc)
Major capsid	0.76	0.76	0.92	0.92	0.83	0.83	2456	0.917	0.918
Minor capsid	0.08	-	0.77	-	0.15	-	81 (0)	0.899	-
Baseplate	0.69	0.69	0.74	0.83	0.72	0.75	851	0.621	0.72
Major tail	0.56	0.53	0.77	0.80	0.65	0.64	502	0.918	0.91
Minor tail	0.75	0.70	0.82	0.81	0.78	0.75	1070	0.939	0.94
Portal	0.83	0.80	0.81	0.85	0.82	0.82	5261	0.943	0.945
Tail fiber	0.31	0.32	0.76	0.75	0.44	0.45	648	0.861	0.86
Tail sheath	0.96	0.95	0.94	0.93	0.95	0.94	2031	0.986	0.957
Collar	0.61	0.53	0.84	0.80	0.70	0.63	300	0.865	0.85
HTJ	0.56	0.58	0.84	0.85	0.67	0.69	1277	0.933	0.923
Others	0.96	0.96	0.87	0.88	0.91	0.92	33402	0.838	0.838
macro avg	0.64	0.68	0.83	0.84	0.69	0.74	47879 (47798)		
weighted avg	0.90	0.90	0.86	0.87	0.88	0.88	47879 (47798)		

https://doi.org/10.1371/journal.pcbi.1007845.t006

We compared the performance of PhANNs with that of VIRALpro by predicting the function class of each other's test set. Doing this requires us to map our 11 classes onto VIRALpro's 4 (capsid versus not-capsid, tail versus not tail). We decided not to use the PhANNs "collar" or "baseplate" test set as VIRALpro has a hard time classifying them (presumably because it was not trained on those classes). Hence we discarded any of the VIRALpro test proteins that PhANNs predicted as "collar" or "baseplate". "Capsid" in VIRALpro means either "major capsid" or "minor capsid" in PhANNs. "Tail" in VIRALpro means "Major tail", "Minor tail", "Tail fiber" or "Tail sheath" in PhANNs. This transformation makes possible the comparison of the two tools. Results are summarized in **Table 7.** The two tools have similar accuracy, with VIRALpro slightly better at predicting capsid proteins and PhANNs slightly better at predicting tail proteins. It is important to mention that the VIRALpro predictions took several days on a 200+ CPU cluster (it would take a few years on a laptop). A similarly sized test takes less than an hour using the PhANNs server.

The utility of the PhANNs tool is to permit more extensive function predictions of metagenome sequences from phages used for phage therapy (A. Cobian, N. Jacobson, M. Rojas, H. Hamza, R. Rowe, D. Conrad, and A. Segall, et al., work in progress) and to better describe the coding potential of the virome in patients suffering from diseases such as inflammatory bowel disease versus household controls (A. Segall, R. Edwards, A. Cantu, S. Handley, and D. Wang, work in progress). In some cases, phage-associated sequences from isolated viromes have no or very weak functional predictions when using BLAST, RPS-BLAST, or related bioinformatic tools (work in progress). In parallel, we are experimentally validating some of the predicted functions using electron microscopy and X-ray crystallography (S.H. Hung, V. Seguritan, et al., ms. in preparation).

Table 7. Comparison of PhANNs with VIRALpro. Results from using VIRALpro test set in PhANNs and PhANNs test set in VIRALpro.

	PhANNs test set in TAILpro	TAILpro test set in PhANNs	PhANNs test set in CAPSIDpro	CAPSIDpro test set in PhANNs
test set size	10,805	672	15,107	787
precision	0.28	0.77	0.14	0.82
recall	0.79	0.68	0.86	0.32
accuracy	0.80	0.82	0.70	0.67
F1-score	0.42	0.72	0.25	0.46

https://doi.org/10.1371/journal.pcbi.1007845.t007

The performance of any machine learning system is limited by the availability and cost of training examples [14]. Invariably, top performing image and audio classification systems must augment their training data with synthetic examples created by applying semantically orthogonal transformations to the training set (i.e., slightly rotating or distorting an image, or adding background noise to audio) [31,32]. In bioinformatics, the current practice of de-replication moves us in exactly the opposite direction—perfectly good samples cannot be used if their overlap with other samples is too high, leaving only one version of the biostring to use for training, thereby ignoring sequence variations. This despite the fact that biological examples such as protein sequence data are replete with variations from a consensus sequence or motif. Our approach overcomes this failing by using *all* non-redundant data. By splitting the dataset into the training, validation, and test sets after first de-replicating at 40%, we remove even slightly redundant samples and make sure that none of the performance is due to data memorization rather than generalization. Augmenting the training set by expanding the clusters to include all non-redundant samples is the novel idea we have introduced in the present paper as a way of increasing our training set size and hence our accuracy.

Conclusion

ANNs are a powerful tool to classify phage structural proteins when homology-based alignments do not provide useful functional predictions, such as "hypothetical" or "unknown function". This approach will become more accurate as more and better characterized phage structural protein sequences, especially more divergent ones, are experimentally validated and become available for inclusion in our training sets. This method can also be applied to predicting the function of unknown proteins of prophage origin in bacterial genomes. In the future, we plan to expand this approach to more protein classes and to viruses of eukaryotes and archaea.

Supporting information

S1 Table. Side chain groupings.
(XLS)

S1 Fig. Model-specific F_1 score—F_1 scores (harmonic mean of precision and recall) for each side chain model/class combination. All models follow similar trends as to which classes are more or less difficult to classify correctly. Error bars represent the 95% confidence intervals.
(PNG)

S2 Fig. Class-specific F_1 score—F_1 scores (harmonic mean of precision and recall) for each side chain model/class combination. Some classes, such as minor capsid, tail fiber, or minor tail, are harder to classify correctly irrespective of the model used. Error bars represent the 95% confidence intervals.
(PNG)

S3 Fig. Comparison of the validation weighted average F_1-score of three models on the same feature sets—We compared our ANN ensemble trained on 1D-10D sets against a logistic regression trained on the 1D-10D sets and an ANN ensemble trained on the 1d-10d sets (40% dereplication, without cluster expansion—see Methods). The ANN ensembles perform significantly better than the logistic regression. Error bars represent 0.95 confidence intervals.
(PNG)

S4 Fig. Per class comparison of the validation F_1-score of three models on the "tetra-s_sc_tri_p feature" set—In the structural classes, the 1D-10D ANN ensemble performs slightly better than the logistic regression and significantly better than the 1d-10d ANN ensemble. In the "others" class (by far the largest), 1D-10D ANN ensemble performs as well as 1d-10d ANN and better than logistic regression. Error bars represent 0.95 confidence intervals.
(PNG)

Acknowledgments

AMS would like to acknowledge Drs. Sherwood Casjens (University of Utah) and Ian Molineux (University of Texas Austin) for helpful conversations on phage biology.

Author Contributions

Conceptualization: Vito Adrian Cantu, Peter Salamon, Victor Seguritan, Jackson Redfield, Robert A. Edwards, Anca M. Segall.

Data curation: Vito Adrian Cantu, Robert A. Edwards, Anca M. Segall.

Formal analysis: Vito Adrian Cantu, Peter Salamon, Robert A. Edwards.

Funding acquisition: Robert A. Edwards, Anca M. Segall.

Investigation: Vito Adrian Cantu, Anca M. Segall.

Methodology: Vito Adrian Cantu, Peter Salamon, David Salamon, Robert A. Edwards.

Project administration: Anca M. Segall.

Resources: Robert A. Edwards, Anca M. Segall.

Software: Vito Adrian Cantu, Robert A. Edwards.

Supervision: Peter Salamon, Anca M. Segall.

Validation: Vito Adrian Cantu, Peter Salamon.

Visualization: Vito Adrian Cantu.

Writing – original draft: Vito Adrian Cantu, Peter Salamon, Anca M. Segall.

Writing – review & editing: Vito Adrian Cantu, Peter Salamon, David Salamon, Robert A. Edwards, Anca M. Segall.

References

1. Cobián Güemes AG, Youle M, Cantú VA, Felts B, Nulton J, Rohwer F. Viruses as Winners in the Game of Life. Annu Rev Virol. 2016 Sep 29; 3(1):197–214. https://doi.org/10.1146/annurev-virology-100114-054952 PMID: 27741409
2. Waldor MK, Mekalanos JJ. Lysogenic conversion by a filamentous phage encoding *cholera* toxin. Science. 1996 Jun 28; 272(5270):1910–4. https://doi.org/10.1126/science.272.5270.1910 PMID: 8658163
3. Breitbart M, Bonnain C, Malki K, Sawaya NA. Phage puppet masters of the marine microbial realm. Nat Microbiol. 2018 Jul; 3(7):754–66. https://doi.org/10.1038/s41564-018-0166-y PMID: 29867096
4. Frank JA, Lorimer D, Youle M, Witte P, Craig T, Abendroth J, et al. Structure and function of a cyanophage-encoded peptide deformylase. ISME J. 2013 Jun; 7(6):1150–60. https://doi.org/10.1038/ismej.2013.4 PMID: 23407310
5. Knowles B, Silveira CB, Bailey BA, Barott K, Cantu VA, Cobián-Güemes AG, et al. Lytic to temperate switching of viral communities. Nature. 2016 Mar 24; 531(7595):466–70. https://doi.org/10.1038/nature17193 PMID: 26982729

6. Kang HS, McNair K, Cuevas DA, Bailey BA, Segall AM, Edwards RA. Prophage genomics reveals patterns in phage genome organization and replication. bioRxiv. 2017 Mar 7;114819.
7. Edwards RA, Rohwer F. Viral metagenomics. Nat Rev Microbiol. 2005 Jun; 3(6):504–10. https://doi.org/10.1038/nrmicro1163 PMID: 15886693
8. McCallin S, Sacher JC, Zheng J, Chan BK. Current State of Compassionate Phage Therapy. Viruses. 2019 Apr; 11(4):343. https://doi.org/10.3390/v11040343 PMID: 31013833
9. Hesse S, Adhya S. Phage Therapy in the Twenty-First Century: Facing the Decline of the Antibiotic Era; Is It Finally Time for the Age of the Phage? Annu Rev Microbiol. 2019; 73(1):155–74. https://doi.org/10.1146/annurev-micro-090817-062535 PMID: 31185183
10. Seguritan V, Alves N Jr., Arnoult M, Raymond A, Lorimer D, Burgin AB Jr., et al. Artificial Neural Networks Trained to Detect Viral and Phage Structural Proteins. PLoS Comput Biol. 2012; 8(8). https://doi.org/10.1371/journal.pcbi.1002657 PMID: 22927809
11. Galiez C, Magnan CN, Coste F, Baldi P. VIRALpro: A tool to identify viral capsid and tail sequences. Bioinformatics. 2016; 32(9):1405–7. https://doi.org/10.1093/bioinformatics/btv727 PMID: 26733451
12. Csáji BC. Approximation with Artificial Neural Networks. 2001; 45.
13. Veesler D, Cambillau C. A common evolutionary origin for tailed bacteriophage functional modules and bacterial machineries. Micr Mol Biol Rev. 2011; 75(3):423–33. https://doi.org/10.1128/MMBR.00014-11 PMID: 21885679
14. Halevy A, Norvig P, Pereira F. The Unreasonable Effectiveness of Data. IEEE Intell Syst. 2009 Mar; 24 (2):8–12.
15. Wattam AR, Davis JJ, Assaf R, Boisvert S, Brettin T, Bun C, et al. Improvements to PATRIC, the all-bacterial Bioinformatics Database and Analysis Resource Center. Nucleic Acids Res. 2017 04; 45(D1): D535–42. https://doi.org/10.1093/nar/gkw1017 PMID: 27899627
16. McNair K, Zhou C, Dinsdale EA, Souza B, Edwards RA. PHANOTATE: a novel approach to gene identification in phage genomes. Bioinforma Oxf Engl. 2019 Apr 25;
17. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinforma Oxf Engl. 2006 Jul 1; 22(13):1658–9.
18. Guruprasad K, Reddy BVB, Pandit MW. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. Protein Eng Des Sel. 1990 Dec 1; 4(2):155–61.
19. Lobry JR, Gautier C. Hydrophobicity, expressivity and aromaticity are the major trends of amino-acid usage in 999 Escherichia coli chromosome-encoded genes. Nucleic Acids Res. 1994 Aug 11; 22 (15):3174–80. https://doi.org/10.1093/nar/22.15.3174 PMID: 8065933
20. Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol. 1982 May 5; 157(1):105–32. https://doi.org/10.1016/0022-2836(82)90515-0 PMID: 7108955
21. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinforma Oxf Engl. 2009 Jun 1; 25 (11):1422–3. https://doi.org/10.1093/bioinformatics/btp163 PMID: 19304878
22. Chollet F, others. Keras [Internet]. 2015. Available from: https://keras.io
23. Abadi Martín, Agarwal Ashish, Barham Paul, Brevdo Eugene, Chen Zhifeng, Citro Craig, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Internet]. 2015. Available from: https://www.tensorflow.org/
24. Drexler K, Dannull J, Hindennach I, Mutschler B, Henning U. Single mutations in a gene for a tail fiber component of an Escherichia coli phage can cause an extension from a protein to a carbohydrate as a receptor. J Mol Biol. 1991 Jun 20; 219(4):655–63. https://doi.org/10.1016/0022-2836(91)90662-p PMID: 1829115
25. Desplats C, Krisch HM. The diversity and evolution of the T4-type bacteriophages. Res Microbiol. 2003 May; 154(4):259–67.
26. Medhekar B, Miller JF. Diversity-generating retroelements. Curr Opin Microbiol. 2007 Aug; 10(4):388–95. https://doi.org/10.1016/j.mib.2007.06.004 PMID: 17703991
27. Ciezki K, Murfin K, Goodrich-Blair H, Stock SP, Forst S. R-type bacteriocins in related strains of *Xenorhabdus bovienii*: Xenorhabdicin tail fiber modularity and contribution to competitiveness. FEMS Microbiol Lett. 2017; 364(1). https://doi.org/10.1093/femsle/fnw235 PMID: 27737947
28. Akusobi C, Chan BK, Williams ESCP, Wertz JE, Turner PE. Parallel Evolution of Host-Attachment Proteins in Phage PP01 Populations Adapting to Escherichia coli O157:H7. Pharm Basel Switz. 2018 Jun 20; 11(2). https://doi.org/10.3390/ph11020060 PMID: 29925767

29. Benler S, Cobián-Güemes AG, McNair K, Hung S-H, Levi K, Edwards R, et al. A diversity-generating retroelement encoded by a globally ubiquitous *Bacteroides* phage. Microbiome. 2018 23; 6(1):191. https://doi.org/10.1186/s40168-018-0573-6 PMID: 30352623

30. Jordan TC, Burnett SH, Carson S, Caruso SM, Clase K, DeJong RJ, et al. A Broadly Implementable Research Course in Phage Discovery and Genomics for First-Year Undergraduate Students. mBio [Internet]. 2014 Feb 4 [cited 2019 Nov 13]; 5(1). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3950523/ https://doi.org/10.1128/mBio.01051-13 PMID: 24496795

31. Kanda N, Takeda R, Obuchi Y. Elastic spectral distortion for low resource speech recognition with deep neural networks. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. 2013. p. 309–14.

32. Ciregan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2012. p. 3642–9.

33. Feng P-M, Ding H, Chen W, Lin H. Naïve bayes classifier with feature selection to identify phage virion proteins. Comput Math Methods Med. 2013;2013. https://doi.org/10.1155/2013/530696 PMID: 23762187

34. Zhang L, Zhang C, Gao R, Yang R. An ensemble method to distinguish bacteriophage virion from non-virion proteins based on protein sequence characteristics. Int J Mol Sci. 2015; 16(9):21734–58. https://doi.org/10.3390/ijms160921734 PMID: 26370987

35. Manavalan B, Shin TH, Lee G. PVP-SVM: Sequence-based prediction of phage virion proteins using a support vector machine. Front Microbiol. 2018; 9(MAR). https://doi.org/10.3389/fmicb.2018.00476 PMID: 29616000

4.3 DISCUSSION

Given the length and scope constrains of the PhANNs paper a few interesting data and discussion were left out. Some of them answer a specific concern, like whether ANNs are better than a logistic regression for this specific application.

4.3.1 Logistic regression

Logistic regression is a ML technique that models the probability that an observation belongs to a particular class. While the method doesn't provide a classification itself, one can easily be derived by thresholding the probabilities (much in the same way PhANNs thresholds sigmoid's outputs). Logistic regression is easier to implement and generates smaller models than ANN. As opposed to ANNs, logistic regressions also produces interpretable models, that is we can interpret directly the impact of its parameters on the outcome. If the performance is better than the ANNs, a Logistic Regression model would be preferred.

To test which model performs better, a logistic regression was trained and evaluated using 10-fold cross validation on the same feature sets. Figure 4.3 shows that ANN (even the ones trained in a reduced set) performs better than logistic regression for all feature sets. In particular, for the feature set "tetra_sc_tri_p" the mean F_1 score is 0.86 for the logistic model and 0.90 for the ANNs trained on the expanded set (and also 0.90 for the reduced set).

4.3.2 Expanded cluster

As shown in figure 4.3, the model trained on the expanded clusters performs about the same as the one trained on the dereplicated proteins for most feature sets. The ANN trained on expanded sets however, performs much better at classifying most class (see Figure 4.4). The mean F_1 score is close because the "others" class is so large compared to the structural classes.

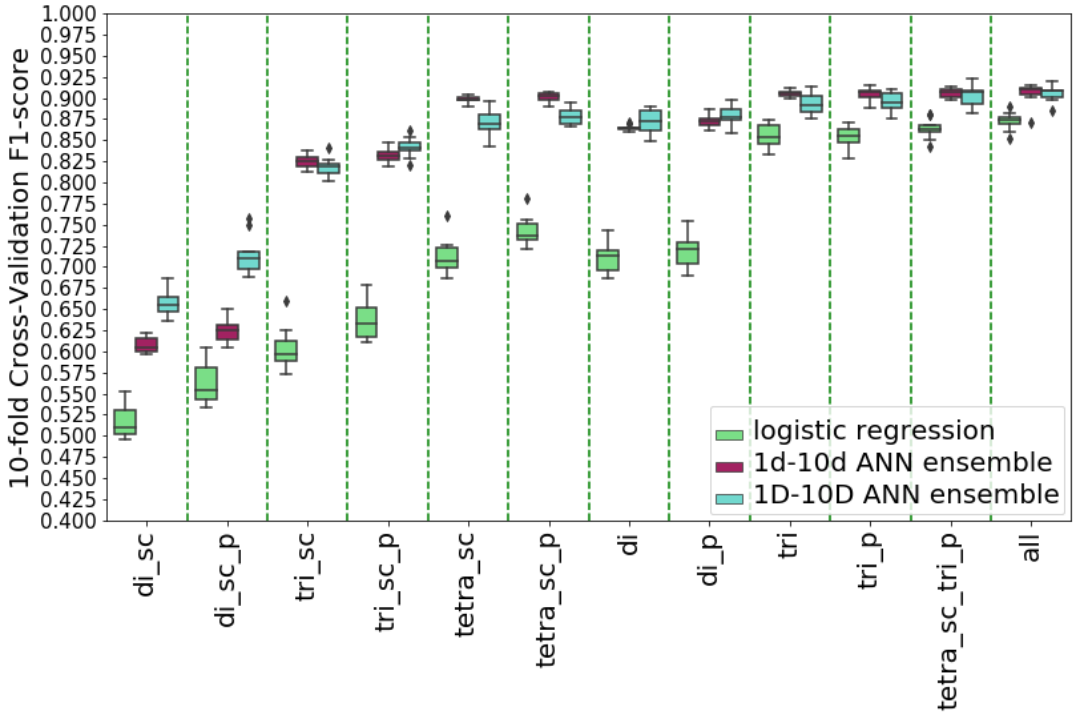


Figure 4.3. Comparison of a Logistic Regression, an ANN trained using the reduced set and a ANN trained using the expanded set

4.3.3 Model size

A concern while using ANNs is that one can easily define a network architecture that generate a model that is impractically large, both regarding the number of trainable parameters and the size of the model files. Using the architecture defined in the PhANNs paper[14] (two hidden layers of 200 neurons each) the model using only tripeptides has 65,650,611 trainable parameters and uses 751.35 Mb of disk space (see table 4.2). Grouping the amino acids by side chains (Table 4.1) reduces the number of trainable parameters to 229,203 and the file size to 2.66 Mb.

It is important to note that to do predictions in a efficient manner, the ten models of an ensemble need to be loaded in memory and that most consumer grade GPUs only have a few Gb of video RAM. (The models can be run from normal RAM, but performance will be affected.) Using side chains of the tetrapeptides generates a model with 6,290,013 trainable

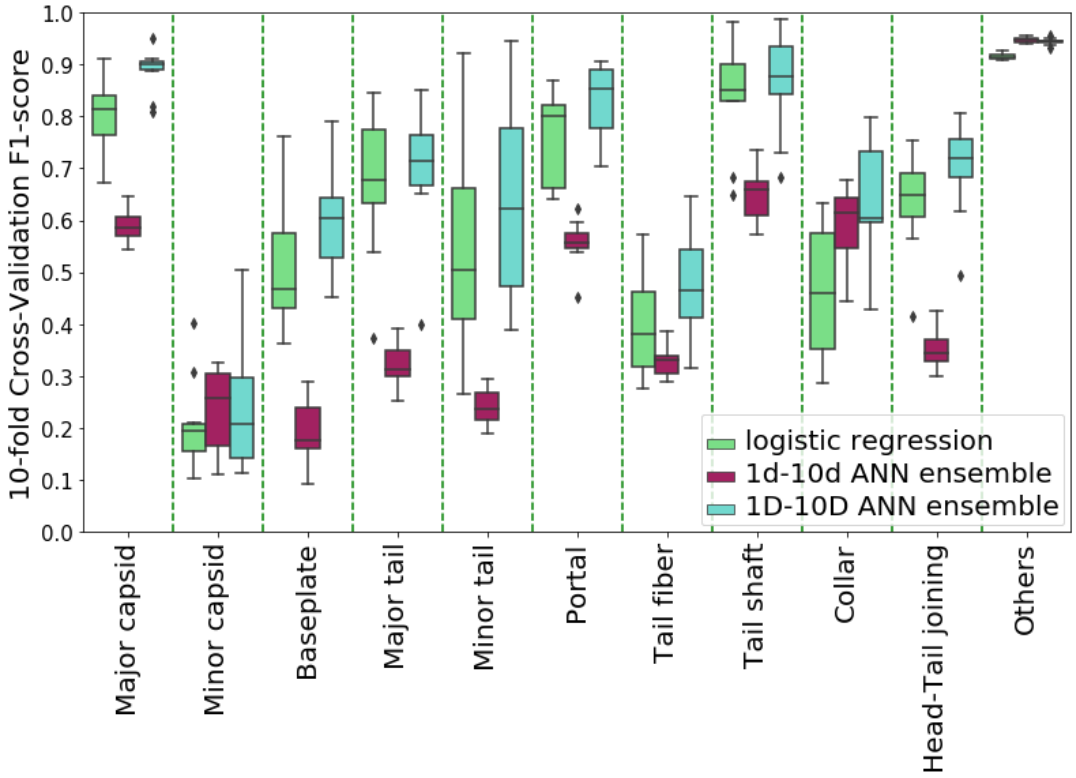


Figure 4.4. "tetra_sc_tri_p" class performance comparison

parameters and file size of 72.02 Mb and the model has a performance comparable to the tripeptide one (see Figure 4.3. All in all, if memory was a limiting factor the use of side-chain models would be preferred.

Table 4.1. Side chain groups

Hydrophobic	A,I,L,M,V
Hydrophylic	N,Q,S,T
Small turn	G,P
disulfide	C
Positive charge	H,K,R
Negative charge	D,E
Aromatic	F,W,Y

Table 4.2. Model size

Feature set	Trainable parameters	File size (Mb)
di_sc	54,861	0.67
di_sc_p	57,317	0.7
tri_sc	229,203	2.66
tri_sc_p	236,363	2.75
tetra_sc	6,290,013	72.02
tetra_sc_p	6,330,101	72.48
di	283,011	3.28
di_p	291,083	3.37
tri	65,650,611	751.35
tri_p	65,780,283	752.84
tetra_sc_tri_p	110,482,101	1264.41
all	127,756,413	1462.1

4.3.4 Web server

PhANNs use of more complex models is aided by the fact that the main mode of usage is through the web server. The web server provides both a suitable environment to run prediction (with enough RAM and the right software installed) and an easy to use graphical user interface. This allows PhANNs to attract users even if they don't have the right equipment or technical know-how. As has been said: "One of the requirements for a successful scientific tool is its availability. Developing a functional web service, however, is usually considered a mundane and ungratifying task, and quite often neglected."[81] Just as the changing times have forced every molecular biologist to be a little of a bioinformatician, they are forcing bioinformaticians to be a little of a web developer.

BIBLIOGRAPHY

[1] *The Nobel Prize in Chemistry 1980*.

[2] L. M. AALL-JILEK, *Epilepsy in the Wapogoro Tribe in Tanganyika1*, Acta Psychiatrica Scandinavica, 41 (1965), pp. 57–86.

[3] B. ALBERTS, *Molecular biology of the cell*, Garland Science, Taylor and Francis Group, New York, NY, sixth edition ed., 2015.

[4] J. ALNEBERG, B. S. BJARNASON, I. DE BRUIJN, M. SCHIRMER, J. QUICK, U. Z. IJAZ, L. LAHTI, N. J. LOMAN, A. F. ANDERSSON, AND C. QUINCE, *Binning metagenomic contigs by coverage and composition*, Nature Methods, 11 (2014), pp. 1144–1146. Number: 11 Publisher: Nature Publishing Group.

[5] R. I. AMANN, W. LUDWIG, AND K. H. SCHLEIFER, *Phylogenetic identification and in situ detection of individual microbial cells without cultivation.*, Microbiological Reviews, 59 (1995), pp. 143–169.

[6] S. ANDREWS, *FastQC A Quality Control tool for High Throughput Sequence Data*, 2010.

[7] A. BANKEVICH, S. NURK, D. ANTIPOV, A. A. GUREVICH, M. DVORKIN, A. S. KULIKOV, V. M. LESIN, S. I. NIKOLENKO, S. PHAM, A. D. PRJIBELSKI, A. V. PYSHKIN, A. V. SIROTKIN, N. VYAHHI, G. TESLER, M. A. ALEKSEYEV, AND P. A. PEVZNER, *SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing*, Journal of Computational Biology, 19 (2012), pp. 455–477.

[8] A. L. BAZINET AND M. P. CUMMINGS, *A comparative evaluation of sequence classification programs*, BMC Bioinformatics, 13 (2012), p. 92.

[9] B. H. BLOOM, *Space/Time Trade-offs in Hash Coding with Allowable Errors*, Commun. ACM, 13 (1970), pp. 422–426.

[10] T. BOLUKBASI, K.-W. CHANG, J. Y. ZOU, V. SALIGRAMA, AND A. T. KALAI, *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings*, p. 9.

[11] R. BOOM, C. J. SOL, M. M. SALIMANS, C. L. JANSEN, P. M. WERTHEIM-VAN DILLEN, AND J. VAN DER NOORDAA, *Rapid and simple method for purification of nucleic acids*, Journal of Clinical Microbiology, 28 (1990), pp. 495–503.

[12] L. BREIMAN, *Random Forests*, Machine Learning, 45 (2001), pp. 5–32.

[13] V. A. CANTU, J. SADURAL, AND R. EDWARDS, *PRINSEQ++, a multi-threaded tool*

for fast and efficient quality control and preprocessing of sequencing datasets, Technical Report e27553v1, PeerJ Inc., Feb. 2019.

[14] V. A. CANTU, P. SALAMON, V. SEGURITAN, J. REDFIELD, D. SALAMON, R. A. EDWARDS, AND A. M. SEGALL, *PhANNs, a fast and accurate tool and web server to classify phage structural proteins*, PLOS Computational Biology, 16 (2020), p. e1007845. Publisher: Public Library of Science.

[15] M. CANUTI, N. J. M. VAN BEVEREN, S. M. JAZAERI FARSANI, M. DE VRIES, M. DEIJS, M. F. JEBBINK, H. L. ZAAIJER, B. D. C. VAN SCHAIK, A. H. C. VAN KAMPEN, A. C. VAN DER KUYL, L. DE HAAN, J. G. STOROSUM, AND L. VAN DER HOEK, *Viral metagenomics in drug-naïve, first-onset schizophrenia patients with prominent negative symptoms*, Psychiatry Research, 229 (2015), pp. 678–684.

[16] N. CHAUDHARY, A. K. SHARMA, P. AGARWAL, A. GUPTA, AND V. K. SHARMA, *16S Classifier: A Tool for Fast and Accurate Taxonomic Classification of 16S rRNA Hypervariable Regions in Metagenomic Datasets*, PLOS ONE, 10 (2015), p. e0116106. Publisher: Public Library of Science.

[17] R. COLEBUNDERS, A. HENDY, J. L. MOKILI, J. F. WAMALA, J. KADUCU, L. KUR, F. TEPAGE, M. MANDRO, G. MUCINYA, G. MAMBANDU, M. Y. KOMBA, J. L. LUMALIZA, M. VAN OIJEN, AND A. LAUDISOIT, *Nodding syndrome and epilepsy in onchocerciasis endemic regions: comparing preliminary observations from South Sudan and the Democratic Republic of the Congo with data from Uganda*, BMC Research Notes, 9 (2016).

[18] R. COLEBUNDERS, A. HENDY, M. NANYUNJA, J. F. WAMALA, AND M. VAN OIJEN, *Nodding syndrome—a new hypothesis and new direction for research*, International Journal of Infectious Diseases, 27 (2014), pp. 74–77.

[19] R. COLEBUNDERS, M. MANDRO, J. L. MOKILI, G. MUCINYA, G. MAMBANDU, K. PFARR, I. REITER-OWONA, A. HOERAUF, F. TEPAGE, B. LEVICK, M. BEGON, AND A. LAUDISOIT, *Risk factors for epilepsy in Bas-Uélé Province, Democratic Republic of the Congo: a case–control study*, International Journal of Infectious Diseases, 49 (2016), pp. 1–8.

[20] F. S. COLLINS, *Implications of the Human Genome Project for Medical Science*, JAMA, 285 (2001), p. 540.

[21] M. COTTEN, B. OUDE MUNNINK, M. CANUTI, M. DEIJS, S. J. WATSON, P. KELLAM, AND L. VAN DER HOEK, *Full genome virus detection in fecal samples using sensitive nucleic acid preparation, deep sequencing, and a novel iterative sequence classification algorithm*, PloS One, 9 (2014), p. e93269.

[22] B. C. CSÁJI, *Approximation with Artificial Neural Networks*, (2001), p. 45.

[23] A. E. DARLING, G. JOSPIN, E. LOWE, F. A. M. IV, H. M. BIK, AND J. A. EISEN, *PhyloSift: phylogenetic analysis of genomes and metagenomes*, PeerJ, 2 (2014), p. e243. Publisher: PeerJ Inc.

[24] M. DE VRIES, M. DEIJS, M. CANUTI, B. D. C. VAN SCHAIK, N. R. FARIA, M. D. B. VAN DE GARDE, L. C. M. JACHIMOWSKI, M. F. JEBBINK, M. JAKOBS, A. C. M. LUYF, F. E. J. COENJAERTS, E. C. J. CLAAS, R. MOLENKAMP, S. M. KOEKKOEK, C. LAMMENS, F. LEUS, H. GOOSSENS, M. IEVEN, F. BAAS, AND L. VAN DER HOEK, *A sensitive assay for virus discovery in respiratory clinical samples*, PloS One, 6 (2011), p. e16118.

[25] A. Y. DEBRAH, S. SPECHT, U. KLARMANN-SCHULZ, L. BATSA, S. MAND, Y. MARFO-DEBREKYEI, R. FIMMERS, B. DUBBEN, A. KWARTENG, M. OSEI-ATWENEBOANA, D. BOAKYE, A. RICCHIUTO, M. BÜTTNER, O. ADJEI, C. D. MACKENZIE, AND A. HOERAUF, *Doxycycline Leads to Sterility and Enhanced Killing of Female Onchocerca volvulus Worms in an Area With Persistent Microfilaridermia After Repeated Ivermectin Treatment: A Randomized, Placebo-Controlled, Double-Blind Trial*, Clinical Infectious Diseases: An Official Publication of the Infectious Diseases Society of America, 61 (2015), pp. 517–526.

[26] T. Z. DESANTIS, P. HUGENHOLTZ, N. LARSEN, M. ROJAS, E. L. BRODIE, K. KELLER, T. HUBER, D. DALEVI, P. HU, AND G. L. ANDERSEN, *Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB*, Applied and Environmental Microbiology, 72 (2006), pp. 5069–5072.

[27] X. DING, F. CHENG, C. CAO, AND X. SUN, *DectICO: an alignment-free supervised metagenomic classification method based on feature extraction and dynamic selection*, BMC Bioinformatics, 16 (2015), p. 323.

[28] S. F. DOWELL, J. J. SEJVAR, L. RIEK, K. A. VANDEMAELE, M. LAMUNU, A. C. KUESEL, E. SCHMUTZHARD, W. MATUJA, S. BUNGA, J. FOLTZ, T. B. NUTMAN, A. S. WINKLER, AND A. K. MBONYE, *Nodding Syndrome*, Emerging Infectious Diseases, 19 (2013), pp. 1374–1373.

[29] B. E. DUTILH, R. SCHMIEDER, J. NULTON, B. FELTS, P. SALAMON, R. A. EDWARDS, AND J. L. MOKILI, *Reference-independent comparative metagenomics using cross-assembly: crAss*, Bioinformatics, 28 (2012), pp. 3225–3231.

[30] R. A. EDWARDS AND F. ROHWER, *Viral metagenomics*, Nature Reviews Microbiology, 3 (2005), pp. 504–510.

[31] D. ENDOH, T. MIZUTANI, R. KIRISAWA, Y. MAKI, H. SAITO, Y. KON, S. MORIKAWA, AND M. HAYASHI, *Species-independent detection of RNA virus by representational difference analysis using non-ribosomal hexanucleotides for reverse transcription*, Nucleic Acids Research, 33 (2005), p. e65.

[32] J. P. EVANS, *Recreational genomics; what's in it for you?*, Genetics in medicine : official journal of the American College of Medical Genetics, 10 (2008), pp. 709–710.

[33] W. FIERS, R. CONTRERAS, F. DUERINCK, G. HAEGEMAN, D. ISERENTANT, J. MERREGAERT, W. MIN JOU, F. MOLEMANS, A. RAEYMAEKERS, A. VAN DEN BERGHE, G. VOLCKAERT, AND M. YSEBAERT, *Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene*, Nature, 260 (1976), pp. 500–507. Number: 5551 Publisher: Nature Publishing Group.

[34] J. L. FOLTZ, I. MAKUMBI, J. J. SEJVAR, M. MALIMBO, R. NDYOMUGYENYI, A. D. ATAI-OMORUTO, L. N. ALEXANDER, B. ABANG, P. MELSTROM, A. M. KAKOOZA, D. OLARA, R. G. DOWNING, T. B. NUTMAN, S. F. DOWELL, AND D. K. W. LWAMAFA, *An Epidemiologic Investigation of Potential Risk Factors for Nodding Syndrome in Kitgum District, Uganda*, PLoS ONE, 8 (2013).

[35] J. GANS, M. WOLINSKY, AND J. DUNBAR, *Computational Improvements Reveal Great Bacterial Diversity and High Metal Toxicity in Soil*, Science, 309 (2005), pp. 1387–1390. Publisher: American Association for the Advancement of Science Section: Report.

[36] P. W. GOLDBERG, *When Can Two Unsupervised Learners Achieve PAC Separation?*, in Computational Learning Theory, D. Helmbold and B. Williamson, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2001, Springer, pp. 303–319.

[37] V. GOMEZ-ALVAREZ, T. K. TEAL, AND T. M. SCHMIDT, *Systematic artifacts in metagenomes from complex microbial communities*, The ISME Journal, 3 (2009), pp. 1314–1317.

[38] J. C. H. M. P. D. G. W. GROUP, *Evaluation of 16S rDNA-Based Community Profiling for Human Microbiome Research*, PLOS ONE, 7 (2012), p. e39315. Publisher: Public Library of Science.

[39] A. GUREVICH, V. SAVELIEV, N. VYAHHI, AND G. TESLER, *QUAST: quality assessment tool for genome assemblies*, Bioinformatics, 29 (2013), pp. 1072–1075.

[40] A. G. C. GÜEMES, M. YOULE, V. A. CANTÚ, B. FELTS, J. NULTON, AND F. ROHWER, *Viruses as Winners in the Game of Life*, Annual Review of Virology, 3 (2016), pp. 197–214.

[41] M. HALDAR, *How much training data do you need?*, May 2019.

[42] A. HALEVY, P. NORVIG, AND F. PEREIRA, *The Unreasonable Effectiveness of Data*, IEEE Intelligent Systems, 24 (2009), pp. 8–12. Conference Name: IEEE Intelligent Systems.

[43] J. HANDELSMAN, M. R. RONDON, S. F. BRADY, J. CLARDY, AND R. M. GOODMAN, *Molecular biological access to the chemistry of unknown soil microbes: a*

new frontier for natural products, Chemistry & Biology, 5 (1998), pp. R245–R249.

[44] T. B. HIGAZI, A. FILIANO, C. R. KATHOLI, Y. DADZIE, J. H. REMME, AND T. R. UNNASCH, *Wolbachia endosymbiont levels in severe and mild strains of Onchocerca volvulus*, Molecular and Biochemical Parasitology, 141 (2005), pp. 109–112.

[45] K. J. HOFF, T. LINGNER, P. MEINICKE, AND M. TECH, *Orphelia: predicting genes in metagenomic sequencing reads*, Nucleic Acids Research, 37 (2009), pp. W101–105.

[46] D. H. HUSON, S. BEIER, I. FLADE, A. GÓRSKA, M. EL-HADIDI, S. MITRA, H.-J. RUSCHEWEYH, AND R. TAPPU, *MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data*, PLOS Computational Biology, 12 (2016), p. e1004957. Publisher: Public Library of Science.

[47] R. IDRO, R. ANGUZU, R. OGWANG, P. AKUN, C. ABBO, A. D. MWAKA, B. OPAR, P. NAKAMYA, M. TAYLOR, A. ELLIOTT, A. VINCENT, C. NEWTON, AND K. MARSH, *Doxycycline for the treatment of nodding syndrome (DONS); the study protocol of a phase II randomised controlled trial*, BMC Neurology, 19 (2019), p. 35.

[48] R. IDRO, R. O. OPOKA, H. T. AANYU, A. KAKOOZA-MWESIGE, T. PILOYA-WERE, H. NAMUSOKE, S. B. MUSOKE, J. NALUGYA, P. BANGIRANA, A. D. MWAKA, S. WHITE, K. CHONG, A. D. ATAI-OMORUTO, E. MWOROZI, J. NANKUNDA, S. KIGULI, J. R. ACENG, AND J. K. TUMWINE, *Nodding syndrome in Ugandan children—clinical features, brain imaging and complications: a case series*, BMJ Open, 3 (2013), p. e002540.

[49] I. ILUMINA, *Sequencing Platforms | Compare NGS platform applications & specifications - https://www.illumina.com/systems/sequencing-platforms.html*.

[50] B. JIANG, K. SONG, J. REN, M. DENG, F. SUN, AND X. ZHANG, *Comparison of metagenomic samples using sequence signatures*, BMC Genomics, 13 (2012), p. 730.

[51] A. D. JOHNSON, *An extended IUPAC nomenclature code for polymorphic nucleic acids*, Bioinformatics, 26 (2010), pp. 1386–1389.

[52] T. P. JOHNSON, R. TYAGI, P. R. LEE, M.-H. LEE, K. R. JOHNSON, J. KOWALAK, A. ELKAHLOUN, M. MEDYNETS, A. HATEGAN, J. KUBOFCIK, J. SEJVAR, J. RATTO, S. BUNGA, I. MAKUMBI, J. R. ACENG, T. B. NUTMAN, S. F. DOWELL, AND A. NATH, *Nodding syndrome may be an autoimmune reaction to the parasitic worm Onchocerca volvulus*, Science Translational Medicine, 9 (2017), p. eaaf6953.

[53] S. JOHRI, J. SOLANKI, V. A. CANTU, S. R. FELLOWS, R. A. EDWARDS, I. MORENO, A. VYAS, AND E. A. DINSDALE, *'Genome skimming' with the MinION hand-held sequencer identifies CITES-listed shark species in India's exports market*, Scientific Reports, 9 (2019), p. 4476.

[54] S. KARIIN AND C. BURGE, *Dinucleotide relative abundance extremes: a genomic*

signature, Trends in Genetics, 11 (1995), pp. 283–290.

[55] L. KAUFMAN AND P. J. ROUSSEEUW, *Finding groups in data: an introduction to cluster analysis*, Wiley series in probability and mathematical statistics, Wiley, New York, 1990.

[56] W. J. KOZEK AND H. F. MARROQUIN, *Intracytoplasmic bacteria in Onchocerca volvulus*, The American Journal of Tropical Medicine and Hygiene, 26 (1977), pp. 663–678.

[57] E. S. LANDER, L. M. LINTON, B. BIRREN, C. NUSBAUM, M. C. ZODY, J. BALDWIN, K. DEVON, K. DEWAR, M. DOYLE, W. FITZHUGH, R. FUNKE, D. GAGE, K. HARRIS, A. HEAFORD, J. HOWLAND, L. KANN, J. LEHOCZKY, R. LEVINE, P. MCEWAN, K. MCKERNAN, J. MELDRIM, J. P. MESIROV, C. MIRANDA, W. MORRIS, J. NAYLOR, C. RAYMOND, M. ROSETTI, R. SANTOS, A. SHERIDAN, C. SOUGNEZ, N. STANGE-THOMANN, N. STOJANOVIC, A. SUBRAMANIAN, D. WYMAN, J. ROGERS, J. SULSTON, R. AINSCOUGH, S. BECK, D. BENTLEY, J. BURTON, C. CLEE, N. CARTER, A. COULSON, R. DEADMAN, P. DELOUKAS, A. DUNHAM, I. DUNHAM, R. DURBIN, L. FRENCH, D. GRAFHAM, S. GREGORY, T. HUBBARD, S. HUMPHRAY, A. HUNT, M. JONES, C. LLOYD, A. MCMURRAY, L. MATTHEWS, S. MERCER, S. MILNE, J. C. MULLIKIN, A. MUNGALL, R. PLUMB, M. ROSS, R. SHOWNKEEN, S. SIMS, R. H. WATERSTON, R. K. WILSON, L. W. HILLIER, J. D. MCPHERSON, M. A. MARRA, E. R. MARDIS, L. A. FULTON, A. T. CHINWALLA, K. H. PEPIN, W. R. GISH, S. L. CHISSOE, M. C. WENDL, K. D. DELEHAUNTY, T. L. MINER, A. DELEHAUNTY, J. B. KRAMER, L. L. COOK, R. S. FULTON, D. L. JOHNSON, P. J. MINX, S. W. CLIFTON, T. HAWKINS, E. BRANSCOMB, P. PREDKI, P. RICHARDSON, S. WENNING, T. SLEZAK, N. DOGGETT, J.-F. CHENG, A. OLSEN, S. LUCAS, C. ELKIN, E. UBERBACHER, M. FRAZIER, R. A. GIBBS, D. M. MUZNY, S. E. SCHERER, J. B. BOUCK, E. J. SODERGREN, K. C. WORLEY, C. M. RIVES, J. H. GORRELL, M. L. METZKER, S. L. NAYLOR, R. S. KUCHERLAPATI, D. L. NELSON, G. M. WEINSTOCK, Y. SAKAKI, A. FUJIYAMA, M. HATTORI, T. YADA, A. TOYODA, T. ITOH, C. KAWAGOE, H. WATANABE, Y. TOTOKI, T. TAYLOR, J. WEISSENBACH, R. HEILIG, W. SAURIN, F. ARTIGUENAVE, P. BROTTIER, T. BRULS, E. PELLETIER, C. ROBERT, P. WINCKER, A. ROSENTHAL, M. PLATZER, G. NYAKATURA, S. TAUDIEN, A. RUMP, D. R. SMITH, L. DOUCETTE-STAMM, M. RUBENFIELD, K. WEINSTOCK, H. M. LEE, J. DUBOIS, H. YANG, J. YU, J. WANG, G. HUANG, J. GU, L. HOOD, L. ROWEN, A. MADAN, S. QIN, R. W. DAVIS, N. A. FEDERSPIEL, A. P. ABOLA, M. J. PROCTOR, B. A. ROE, F. CHEN, H. PAN, J. RAMSER, H. LEHRACH, R. REINHARDT, W. R. MCCOMBIE, M. DE LA BASTIDE, N. DEDHIA, H. BLÖCKER, K. HORNISCHER, G. NORDSIEK, R. AGARWALA, L. ARAVIND, J. A. BAILEY, A. BATEMAN, S. BATZOGLOU, E. BIRNEY, P. BORK, D. G. BROWN, C. B. BURGE, L. CERUTTI, H.-C. CHEN, D. CHURCH, M. CLAMP, R. R. COPLEY, T. DOERKS, S. R. EDDY, E. E. EICHLER, T. S. FUREY, J. GALAGAN, J. G. R. GILBERT, C. HARMON, Y. HAYASHIZAKI,

D. HAUSSLER, H. HERMJAKOB, K. HOKAMP, W. JANG, L. S. JOHNSON, T. A. JONES, S. KASIF, A. KASPRYZK, S. KENNEDY, W. J. KENT, P. KITTS, E. V. KOONIN, I. KORF, D. KULP, D. LANCET, T. M. LOWE, A. MCLYSAGHT, T. MIKKELSEN, J. V. MORAN, N. MULDER, V. J. POLLARA, C. P. PONTING, G. SCHULER, J. SCHULTZ, G. SLATER, A. F. A. SMIT, E. STUPKA, J. SZUSTAKOWKI, D. THIERRY-MIEG, J. THIERRY-MIEG, L. WAGNER, J. WALLIS, R. WHEELER, A. WILLIAMS, Y. I. WOLF, K. H. WOLFE, S.-P. YANG, R.-F. YEH, F. COLLINS, M. S. GUYER, J. PETERSON, A. FELSENFELD, K. A. WETTERSTRAND, R. M. MYERS, J. SCHMUTZ, M. DICKSON, J. GRIMWOOD, D. R. COX, M. V. OLSON, R. KAUL, C. RAYMOND, N. SHIMIZU, K. KAWASAKI, S. MINOSHIMA, G. A. EVANS, M. ATHANASIOU, R. SCHULTZ, A. PATRINOS, M. J. MORGAN, INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM, C. F. G. R. WHITEHEAD INSTITUTE FOR BIOMEDICAL RESEARCH, THE SANGER CENTRE:, WASHINGTON UNIVERSITY GENOME SEQUENCING CENTER, US DOE JOINT GENOME INSTITUTE:, BAYLOR COLLEGE OF MEDICINE HUMAN GENOME SEQUENCING CENTER:, RIKEN GENOMIC SCIENCES CENTER:, GENOSCOPE AND CNRS UMR-8030:, I. O. M. B. DEPARTMENT OF GENOME ANALYSIS, GTC SEQUENCING CENTER:, BEIJING GENOMICS INSTITUTE/HUMAN GENOME CENTER:, T. I. F. S. B. MULTIMEGABASE SEQUENCING CENTER, STANFORD GENOME TECHNOLOGY CENTER:, UNIVERSITY OF OKLAHOMA'S ADVANCED CENTER FOR GENOME TECHNOLOGY:, MAX PLANCK INSTITUTE FOR MOLECULAR GENETICS:, L. A. H. G. C. COLD SPRING HARBOR LABORATORY, GBF—GERMAN RESEARCH CENTRE FOR BIOTECHNOLOGY:, A. I. I. L. U. O. H. *GENOME ANALYSIS GROUP (LISTED IN ALPHABETICAL ORDER, U. N. I. O. H. SCIENTIFIC MANAGEMENT: NATIONAL HUMAN GENOME RESEARCH INSTITUTE, STANFORD HUMAN GENOME CENTER:, UNIVERSITY OF WASHINGTON GENOME CENTER:, K. U. S. O. M. DEPARTMENT OF MOLECULAR BIOLOGY, UNIVERSITY OF TEXAS SOUTHWESTERN MEDICAL CENTER AT DALLAS:, U. D. O. E. OFFICE OF SCIENCE, AND THE WELLCOME TRUST:, *Initial sequencing and analysis of the human genome*, Nature, 409 (2001), pp. 860–921. Number: 6822 Publisher: Nature Publishing Group.

[58] B. LANGMEAD AND S. L. SALZBERG, *Fast gapped-read alignment with Bowtie 2*, Nature Methods, 9 (2012), pp. 357–359. Number: 4 Publisher: Nature Publishing Group.

[59] W. LI AND A. GODZIK, *Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences*, Bioinformatics (Oxford, England), 22 (2006), pp. 1658–1659.

[60] L. LIU, Y. LI, S. LI, N. HU, Y. HE, R. PONG, D. LIN, L. LU, AND M. LAW, *Comparison of Next-Generation Sequencing Systems*, Journal of Biomedicine and Biotechnology, 2012 (2012).

[61] Z. LIU, W. HSIAO, B. L. CANTAREL, E. F. DRÁBEK, AND C. FRASER-LIGGETT,

Sparse distance-based learning for simultaneous multiclass classification and feature selection of metagenomic data, Bioinformatics (Oxford, England), 27 (2011), pp. 3242–3249.

[62] A. LOPEZ-DEL RIO, M. MARTIN, A. PERERA-LLUNA, AND R. SAIDI, *Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction*, Scientific Reports, 10 (2020), p. 14634. Number: 1 Publisher: Nature Publishing Group.

[63] C. LOZUPONE AND R. KNIGHT, *UniFrac: a New Phylogenetic Method for Comparing Microbial Communities*, Applied and Environmental Microbiology, 71 (2005), pp. 8228–8235.

[64] S. LU, J. WANG, F. CHITSAZ, M. K. DERBYSHIRE, R. C. GEER, N. R. GONZALES, M. GWADZ, D. I. HURWITZ, G. H. MARCHLER, J. S. SONG, N. THANKI, R. A. YAMASHITA, M. YANG, D. ZHANG, C. ZHENG, C. J. LANCZYCKI, AND A. MARCHLER-BAUER, *CDD/SPARCLE: the conserved domain database in 2020*, Nucleic Acids Research, 48 (2020), pp. D265–D268.

[65] W. LUDWIG AND K. H. SCHLEIFER, *Bacterial phylogeny based on 16S and 23S rRNA sequence analysis*, FEMS Microbiology Reviews, 15 (1994), pp. 155–173. Publisher: Oxford Academic.

[66] R. MADUPU, A. RICHTER, R. J. DODSON, L. BRINKAC, D. HARKINS, S. DURKIN, S. SHRIVASTAVA, G. SUTTON, AND D. HAFT, *CharProtDB: a database of experimentally characterized protein annotations*, Nucleic Acids Research, 40 (2012), pp. D237–D241.

[67] B. MANAVALAN, T. SHIN, AND G. LEE, *PVP-SVM: Sequence-based prediction of phage virion proteins using a support vector machine*, Frontiers in Microbiology, 9 (2018).

[68] K. MCNAIR, C. ZHOU, E. A. DINSDALE, B. SOUZA, AND R. A. EDWARDS, *PHANOTATE: a novel approach to gene identification in phage genomes*, Bioinformatics, 35 (2019), pp. 4537–4542. Publisher: Oxford Academic.

[69] S. N. MCNULTY, J. M. FOSTER, M. MITREVA, J. C. D. HOTOPP, J. MARTIN, K. FISCHER, B. WU, P. J. DAVIS, S. KUMAR, N. W. BRATTIG, B. E. SLATKO, G. J. WEIL, AND P. U. FISCHER, *Endosymbiont DNA in Endobacteria-Free Filarial Nematodes Indicates Ancient Horizontal Genetic Transfer*, PLOS ONE, 5 (2010), p. e11029. Publisher: Public Library of Science.

[70] T. M. MITCHELL, *Machine Learning*, McGraw-Hill series in computer science, McGraw-Hill, New York, 1997.

[71] A. MORGULIS, E. M. GERTZ, A. A. SCHÄFFER, AND R. AGARWALA, *A fast and*

symmetric DUST implementation to mask low-complexity DNA sequences, Journal of Computational Biology: A Journal of Computational Molecular Cell Biology, 13 (2006), pp. 1028–1040.

[72] F. NARGESIAN, H. SAMULOWITZ, U. KHURANA, E. B. KHALIL, AND D. TURAGA, *Learning Feature Engineering for Classification*, in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, Aug. 2017, International Joint Conferences on Artificial Intelligence Organization, pp. 2529–2535.

[73] NCBI RESOURCE COORDINATORS, *Database resources of the National Center for Biotechnology Information*, Nucleic Acids Research, 42 (2014), pp. D7–D17. Publisher: Oxford Academic.

[74] M. T. NELSON, C. E. POPE, R. L. MARSH, D. J. WOLTER, E. J. WEISS, K. R. HAGER, A. T. VO, M. J. BRITTNACHER, M. C. RADEY, H. S. HAYDEN, A. ENG, S. I. MILLER, E. BORENSTEIN, AND L. R. HOFFMAN, *Human and Extracellular DNA Depletion for Metagenomic Analysis of Complex Clinical Infection Samples Yields Optimized Viable Microbiome Profiles*, Cell Reports, 26 (2019), pp. 2227–2240.e5.

[75] H. NOGUCHI, J. PARK, AND T. TAKAGI, *MetaGene: prokaryotic gene finding from environmental genome shotgun sequences*, Nucleic Acids Research, 34 (2006), pp. 5623–5630.

[76] Z. OBERMEYER, B. POWERS, C. VOGELI, AND S. MULLAINATHAN, *Dissecting racial bias in an algorithm used to manage the health of populations*, Science, 366 (2019), pp. 447–453. Publisher: American Association for the Advancement of Science Section: Research Article.

[77] A. M. O'HARA AND F. SHANAHAN, *The gut flora as a forgotten organ*, EMBO Reports, 7 (2006), pp. 688–693.

[78] N. R. PACE, D. A. STAHL, D. J. LANE, AND G. J. OLSEN, *The Analysis of Natural Microbial Populations by Ribosomal RNA Sequences*, in Advances in Microbial Ecology, K. C. Marshall, ed., Advances in Microbial Ecology, Springer US, Boston, MA, 1986, pp. 1–55.

[79] D. H. PARKS, N. J. MACDONALD, AND R. G. BEIKO, *Classifying short genomic fragments from novel lineages using composition and homology*, BMC Bioinformatics, 12 (2011), p. 328.

[80] A. PARTOW, *General Purpose Hash Function Algorithms*.

[81] PAWEŁ DANILUK, B. WILCZYŃSKI, AND B. LESYNG, *WeBIAS: a web server for publishing bioinformatics applications*, BMC Research Notes, 8 (2015), p. 628.

[82] A. PAYNE, N. HOLMES, V. RAKYAN, AND M. LOOSE, *Whale watching with BulkVis:*

A graphical viewer for Oxford Nanopore bulk fast5 files, bioRxiv, (2018), p. 312256. Publisher: Cold Spring Harbor Laboratory Section: New Results.

[83] R. P. N. RAO, *Probabilistic Analysis of an Ancient Undeciphered Script*, Computer, 43 (2010), pp. 76–80.

[84] M. RHO, H. TANG, AND Y. YE, *FragGeneScan: predicting genes in short and error-prone reads*, Nucleic Acids Research, 38 (2010), p. e191.

[85] A. S, B. BA, S. P, A. RK, AND E. RA, *Applying Shannon's information theory to bacterial and phage genomes and metagenomes.*, Scientific Reports, 3 (2013), pp. 1033–1033.

[86] F. SANGER, G. M. AIR, B. G. BARRELL, N. L. BROWN, A. R. COULSON, C. A. FIDDES, C. A. HUTCHISON, P. M. SLOCOMBE, AND M. SMITH, *Nucleotide sequence of bacteriophage phi X174 DNA*, Nature, 265 (1977), pp. 687–695.

[87] P. D. SCHLOSS AND J. HANDELSMAN, *Introducing DOTUR, a Computer Program for Defining Operational Taxonomic Units and Estimating Species Richness*, Applied and Environmental Microbiology, 71 (2005), pp. 1501–1506. Publisher: American Society for Microbiology Section: METHODS.

[88] R. SCHMIEDER AND R. EDWARDS, *Quality control and preprocessing of metagenomic datasets*, Bioinformatics, 27 (2011), pp. 863–864.

[89] J. J. SEJVAR, A. M. KAKOOZA, J. L. FOLTZ, I. MAKUMBI, A. D. ATAI-OMORUTO, M. MALIMBO, R. NDYOMUGYENYI, L. N. ALEXANDER, B. ABANG, R. G. DOWNING, A. EHRENBERG, K. GUILLIAMS, S. HELMERS, P. MELSTROM, D. OLARA, S. PERLMAN, J. RATTO, E. TREVATHAN, A. S. WINKLER, S. F. DOWELL, AND D. LWAMAFA, *Clinical, neurological, and electrophysiological features of nodding syndrome in Kitgum, Uganda: an observational case series*, The Lancet. Neurology, 12 (2013), pp. 166–174.

[90] G. G. Z. SILVA, D. A. CUEVAS, B. E. DUTILH, AND R. A. EDWARDS, *FOCUS: an alignment-free model to identify organisms in metagenomes using non-negative least squares*, PeerJ, 2 (2014), p. e425.

[91] P. SPENCER, K. VANDEMAELE, M. RICHER, V. PALMER, S. CHUNGONG, M. ANKER, Y. AYANA, M. OPOKA, B. KLAUCKE, A. QUARELLO, AND J. TUMWINE, *Nodding syndrome in Mundri county, South Sudan: environmental, nutritional and infectious factors*, African Health Sciences, 13 (2013), pp. 183–204.

[92] M. J. SULLIVAN, N. K. PETTY, AND S. A. BEATSON, *Easyfig: a genome comparison visualizer*, Bioinformatics, 27 (2011), pp. 1009–1010.

[93] Y. SUN, Y. CAI, L. LIU, F. YU, M. L. FARRELL, W. MCKENDREE, AND W. FARMERIE, *ESPRIT: estimating species richness using large collections of 16S*

rRNA pyrosequences, Nucleic Acids Research, 37 (2009), p. e76.

[94] F. TAMAROZZI, A. HALLIDAY, K. GENTIL, A. HOERAUF, E. PEARLMAN, AND M. J. TAYLOR, *Onchocerciasis: the Role of Wolbachia Bacterial Endosymbionts in Parasite Biology, Disease Pathogenesis, and Treatment*, Clinical Microbiology Reviews, 24 (2011), pp. 459–468.

[95] O. TANASEICHUK, J. BORNEMAN, AND T. JIANG, *Phylogeny-based classification of microbial communities*, Bioinformatics (Oxford, England), 30 (2014), pp. 449–456.

[96] H. TEELING, J. WALDMANN, T. LOMBARDOT, M. BAUER, AND F. O. GLÖCKNER, *TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences*, BMC Bioinformatics, 5 (2004), p. 163.

[97] TIN KAM HO, *The random subspace method for constructing decision forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (1998), pp. 832–844.

[98] M. VALERA, Z. GUO, P. KELLY, S. MATZ, V. A. CANTU, A. G. PERCUS, J. D. HYMAN, G. SRINIVASAN, AND H. S. VISWANATHAN, *Machine learning for graph-based representations of three-dimensional discrete fracture networks*, Computational Geosciences, 22 (2018), pp. 695–710.

[99] L. G. VALIANT, *A theory of the learnable*, Communications of the ACM, 27 (1984).

[100] K. VERVIER, P. MAHÉ, M. TOURNOUD, J.-B. VEYRIERAS, AND J.-P. VERT, *Large-scale machine learning for metagenomics sequence classification*, Bioinformatics (Oxford, England), 32 (2016), pp. 1023–1032.

[101] G. VOGEL, *Parasitic worm may trigger mystery nodding syndrome*, Science, 355 (2017), pp. 678–678. Publisher: American Association for the Advancement of Science Section: In Depth.

[102] F. W. V. D. WAALS, J. GOUDSMIT, AND D. C. GAJDUSEK, *See-ee: Clinical Characteristics of Highly Prevalent Seizure Disorders in the Gbawein and Wroughbarh Clan Region of Grand Bassa County, Liberia*, Neuroepidemiology, 2 (1983), pp. 35–44.

[103] Y. WANG, H. C. M. LEUNG, S. M. YIU, AND F. Y. L. CHIN, *MetaCluster 4.0: a novel binning algorithm for NGS reads and huge number of species*, Journal of Computational Biology: A Journal of Computational Molecular Cell Biology, 19 (2012), pp. 241–249.

[104] J. D. WATSON AND F. H. C. CRICK, *Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid*, Nature, 171 (1953), pp. 737–738. Number: 4356 Publisher: Nature Publishing Group.

[105] A. R. WATTAM, J. J. DAVIS, R. ASSAF, S. BOISVERT, T. BRETTIN, C. BUN,

N. CONRAD, E. M. DIETRICH, T. DISZ, J. L. GABBARD, S. GERDES, C. S. HENRY, R. W. KENYON, D. MACHI, C. MAO, E. K. NORDBERG, G. J. OLSEN, D. E. MURPHY-OLSON, R. OLSON, R. OVERBEEK, B. PARRELLO, G. D. PUSCH, M. SHUKLA, V. VONSTEIN, A. WARREN, F. XIA, H. YOO, AND R. L. STEVENS, *Improvements to PATRIC, the all-bacterial Bioinformatics Database and Analysis Resource Center*, Nucleic Acids Research, 45 (2017), pp. D535–D542.

[106] J. J. WERNER, D. KNIGHTS, M. L. GARCIA, N. B. SCALFONE, S. SMITH, K. YARASHESKI, T. A. CUMMINGS, A. R. BEERS, R. KNIGHT, AND L. T. ANGENENT, *Bacterial community structures are unique and resilient in full-scale bioenergy systems*, Proceedings of the National Academy of Sciences of the United States of America, 108 (2011), pp. 4158–4163.

[107] R. R. WICK, L. M. JUDD, AND K. E. HOLT, *Deepbinner: Demultiplexing barcoded Oxford Nanopore reads with deep convolutional neural networks*, PLOS Computational Biology, 14 (2018), p. e1006583. Publisher: Public Library of Science.

[108] R. R. WICK, M. B. SCHULTZ, J. ZOBEL, AND K. E. HOLT, *Bandage: interactive visualization of de novo genome assemblies*, Bioinformatics, 31 (2015), pp. 3350–3352. Publisher: Oxford Academic.

[109] WIKIPEDIA, *FLOPS*, Oct. 2020. Page Version ID: 982617720.

[110] M. R. WILSON, S. N. NACCACHE, E. SAMAYOA, M. BIAGTAN, H. BASHIR, G. YU, S. M. SALAMAT, S. SOMASEKAR, S. FEDERMAN, S. MILLER, R. SOKOLIC, E. GARABEDIAN, F. CANDOTTI, R. H. BUCKLEY, K. D. REED, T. L. MEYER, C. M. SEROOGY, R. GALLOWAY, S. L. HENDERSON, J. E. GERN, J. L. DERISI, AND C. Y. CHIU, *Actionable Diagnosis of Neuroleptospirosis by Next-Generation Sequencing*, New England Journal of Medicine, 370 (2014), pp. 2408–2417.

[111] S. C. WONG, A. GATT, V. STAMATESCU, AND M. D. MCDONNELL, *Understanding data augmentation for classification: when to warp?*, arXiv:1609.08764 [cs], (2016). arXiv: 1609.08764.

[112] W. ZHU, A. LOMSADZE, AND M. BORODOVSKY, *Ab initio gene identification in metagenomic sequences*, Nucleic Acids Research, 38 (2010), p. e132.

APPENDIX A

NODDING SYNDROME SAMPLES

NODDING SYNDROME SAMPLES

The 63 metagenomes sequenced from the Nodding Syndrome samples taken in the June 2014 exploration [19] are summarized in table A.1.

Table A.1. NS samples.

Sample	Raw Reads	Sample Type	Diagnose	Patient
1	743,010	CSF	case	case 1
2	783,458	CSF	case	case 2
3	776,621	CSF	case	case 3
4	702,658	CSF	case	case 4
5	629,487	CSF	case	case 5
6	817,455	CSF	case	case 6
7	688,579	CSF	case	case 7
8	935,173	CSF	case	case 8
9	448,273	CSF	case	case 9
10	72,8997	CSF	case	case 10
11	781,442	CSF	case	case 11
12	814,782	CSF	case	case 12
13	794,087	CSF	case	case 13
14	773,548	CSF	case	case 14
15	737,696	CSF	case	case 15
16	874,566	CSF	case	case 16
17	788,440	CSF	case	case 17
18	784,414	CSF	case	case 18

(table continues)

Table A.1 (Continued)

Sample	Raw Reads	Sample Type	Diagnose	Patient
19	400,416	plasma	case	case 1
20	322,838	plasma	case	case 2
21	505,256	plasma	case	case 3
22	579,633	plasma	case	case 4
23	419,083	plasma	case	case 5
24	639,418	plasma	case	case 6
25	247,951	plasma	case	case 7
26	516,367	plasma	case	case 8
27	758,899	plasma	case	case 9
28	759,494	plasma	case	case 10
29	1,004,518	plasma	case	case 11
30	810,147	plasma	case	case 12
31	748,227	plasma	case	case 13
32	448,661	plasma	case	case 14
33	877,799	plasma	case	case 15
34	766,670	plasma	case	case 16
35	535,364	plasma	case	case 17
36	417,883	plasma	case	case 18
37	738,959	buffy coat	case	case 1
38	1,300,194	buffy coat	case	case 2
39	655,510	buffy coat	case	case 3
40	887,467	buffy coat	case	case 4
41	765,489	buffy coat	case	case 5

(table continues)

Table A.1 (Continued)

Sample	Raw Reads	Sample Type	Diagnose	Patient
42	713,629	buffy coat	case	case 6
43	798,685	buffy coat	case	case 7
44	939,666	buffy coat	case	case 8
45	805,433	buffy coat	case	case 9
46	650,491	buffy coat	case	case 10
47	814,459	buffy coat	case	case 11
48	877,937	buffy coat	case	case 12
49	967,553	buffy coat	case	case 13
50	661,007	buffy coat	case	case 14
51	768,428	buffy coat	case	case 15
52	871,011	buffy coat	case	case 16
53	773,300	buffy coat	case	case 17
54	768,602	buffy coat	case	case 18
55	402,275	plasma	control	control 1
56	500,492	plasma	control	control 2
57	725,053	plasma	control	control 3
58	649,526	plasma	control	control 4
59	574,179	plasma	control	control 5
60	569,901	plasma	control	control 6
61	531,585	plasma	control	control 7
62	664,373	plasma	control	control 8
63	585,990	plasma	control	control 9

Table A.2 shows annotations and scores for the 100 contigs with highest importance acording the the first Random Forest model (Imp1). Contigs are ranked according the their importance in the second RF model (Imp2). The p-value derived from an unpaired t-test is also included.

Table A.2. Top 100 contig stats

Contig	Annotation	Genus	familly	$-log_{10}(p\text{-}value)$	Imp2	Imp1	rank
NODE_1117	Halomonas	Halomonas	Halomonadaceae	3.68	0.0295	0.0024	1
NODE_201	Halomonas sp. HG01	Halomonas	Halomonadaceae	5.11	0.0090	0.0017	2
NODE_183	Halomonas aestuarii	Halomonas	Halomonadaceae	5.30	0.0082	0.0013	3
NODE_15387	No hits	NA	NA	5.11	0.0074	0.0005	4
NODE_180	Halomonas beimenensis	Halomonas	Halomonadaceae	5.46	0.0067	0.0010	5
NODE_170	Halomonas sp. 1513	Halomonas	Halomonadaceae	2.66	0.0053	0.0012	6
NODE_1172	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.26	0.0052	0.0003	7
NODE_65	Halomonas sp. 1513	Halomonas	Halomonadaceae	3.38	0.0044	0.0008	8
NODE_946	Halomonas	Halomonas	Halomonadaceae	1.78	0.0041	0.0013	9
NODE_953	Halomonas beimenensis	Halomonas	Halomonadaceae	5.02	0.0040	0.0004	10
NODE_454	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.71	0.0038	0.0009	11
NODE_1331	Halomonas sp. 1513	Halomonas	Halomonadaceae	4.69	0.0036	0.0007	12
NODE_80	Halomonadaceae	Chromohalobacter	Halomonadaceae	4.46	0.0030	0.0007	13
NODE_386	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	4.74	0.0030	0.0008	14
NODE_38123	Halomonas	Halomonas	Halomonadaceae	4.95	0.0029	0.0003	15
NODE_57	Halomonas aestuarii	Halomonas	Halomonadaceae	4.38	0.0029	0.0010	16

(table continues)

Table A.2 (Continued)

Contig	Annotation	Genus	familly	$-log_{10}(p\text{-}value)$	Imp2	Imp1	rank
NODE_1149	Halomonas sp. 1513	Halomonas	Halomonadaceae	3.33	0.0028	0.0016	17
NODE_738	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.69	0.0027	0.0008	18
NODE_123	Halomonas aestuarii	Halomonas	Halomonadaceae	4.96	0.0026	0.0012	19
NODE_817	Halomonas aestuarii	Halomonas	Halomonadaceae	3.90	0.0025	0.0005	20
NODE_497	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.76	0.0023	0.0009	21
NODE_35188	Lactobacillales	Lactobacillus	Lactobacillaceae	3.89	0.0023	0.0004	22
NODE_51	Gammaproteobacteria	NA	NA	4.58	0.0022	0.0006	23
NODE_451	Halomonadaceae	NA	Halomonadaceae	3.29	0.0021	0.0006	24
NODE_239	Halomonas	Halomonas	Halomonadaceae	3.97	0.0021	0.0006	25
NODE_184	Halomonadaceae	Chromohalobacter	Halomonadaceae	3.71	0.0021	0.0013	26
NODE_269	Halomonas sp. HG01	Halomonas	Halomonadaceae	5.77	0.0016	0.0009	27
NODE_439	Halomonas sp. 1513	Halomonas	Halomonadaceae	4.70	0.0014	0.0005	28
NODE_228	Halomonas	Halomonas	Halomonadaceae	4.01	0.0014	0.0010	29
NODE_34109	Homo sapiens	Homo	Hominidae	1.40	0.0014	0.0003	30
NODE_68	Halomonas	Halomonas	Halomonadaceae	4.27	0.0013	0.0012	31
NODE_587	Halomonas beimenensis	Halomonas	Halomonadaceae	4.27	0.0013	0.0003	32

(table continues)

Table A.2 (Continued)

Contig	Annotation	Genus	familly	$-log_{10}(p\text{-}value)$	Imp2	Imp1	rank
NODE_391	Halomonadaceae	Salinicola	Halomonadaceae	3.98	0.0012	0.0005	33
NODE_653	Halomonas beimenensis	Halomonas	Halomonadaceae	5.61	0.0012	0.0003	34
NODE_285	Halomonadaceae	Chromohalobacter	Halomonadaceae	5.01	0.0011	0.0010	35
NODE_588	Halomonas sp. HG01	Halomonas	Halomonadaceae	5.86	0.0010	0.0007	36
NODE_116	Halomonas sp. 1513	Halomonas	Halomonadaceae	4.85	0.0010	0.0012	37
NODE_192	Bradyrhizobium sp. G22	Bradyrhizobium	Bradyrhizobiaceae	4.42	0.0010	0.0003	38
NODE_343	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	4.95	0.0009	0.0004	39
NODE_220	Halomonas aestuarii	Halomonas	Halomonadaceae	3.77	0.0009	0.0007	40
NODE_81	Halomonas sp. 1513	Halomonas	Halomonadaceae	3.90	0.0008	0.0005	41
NODE_1385	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.08	0.0008	0.0003	42
NODE_216	Halomonadaceae	Chromohalobacter	Halomonadaceae	5.12	0.0008	0.0012	43
NODE_2132	Pseudomonas putida group	Pseudomonas	Pseudomonadaceae	3.84	0.0008	0.0003	44
NODE_578	Chromobacteriaceae	Pseudogulbenkiania	Chromobacteriaceae	5.48	0.0008	0.0006	45
NODE_1360	Pseudomonas fluorescens group	Pseudomonas	Pseudomonadaceae	4.29	0.0007	0.0004	46
NODE_449	Halomonas sp. HG01	Halomonas	Halomonadaceae	5.18	0.0007	0.0005	47
NODE_52	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	3.83	0.0006	0.0004	48

(table continues)

Table A.2 (Continued)

Contig	Annotation	Genus	familly	$-log_{10}(p\text{-}value)$	Imp2	Imp1	rank
NODE_265	Halomonas aestuarii	Halomonas	Halomonadaceae	3.44	0.0006	0.0003	49
NODE_363	Halomonas	Halomonas	Halomonadaceae	4.99	0.0006	0.0011	50
NODE_481	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	3.78	0.0006	0.0003	51
NODE_689	Gammaproteobacteria	Marinobacter	Alteromonadaceae	1.61	0.0005	0.0006	52
NODE_106	Halomonas sp. 1513	Halomonas	Halomonadaceae	4.92	0.0005	0.0007	53
NODE_150	Halomonadaceae	NA	Halomonadaceae	4.82	0.0004	0.0006	54
NODE_1479	Halomonas aestuarii	Halomonas	Halomonadaceae	2.58	0.0004	0.0003	55
NODE_262	Halomonas	Halomonas	Halomonadaceae	4.54	0.0003	0.0009	56
NODE_204	Halomonas sp. HG01	Halomonas	Halomonadaceae	4.59	0.0003	0.0004	57
NODE_389	Halomonadaceae	NA	Halomonadaceae	5.24	0.0003	0.0003	58
NODE_390	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	3.22	0.0003	0.0004	59
NODE_309	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	3.64	0.0003	0.0003	60
NODE_898	Oceanospirillales	Alcanivorax	Alcanivoracaceae	4.23	0.0003	0.0003	61
NODE_230	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	4.64	0.0002	0.0005	62
NODE_252	Halomonadaceae	Chromohalobacter	Halomonadaceae	5.21	0.0002	0.0003	63
NODE_1387	Halomonas aestuarii	Halomonas	Halomonadaceae	1.05	0.0002	0.0005	64

(table continues)

Table A.2 (Continued)

Contig	Annotation	Genus	familly	$-log_{10}(p\text{-}value)$	Imp2	Imp1	rank
NODE_60	Halomonas	Halomonas	Halomonadaceae	4.16	0.0001	0.0007	65
NODE_382	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.82	0.0001	0.0007	66
NODE_12342	Viruses	Pegivirus	Flaviviridae	1.09	0.0001	0.0005	67
NODE_258	Pseudomonadaceae	Azotobacter	Pseudomonadaceae	5.55	0.0001	0.0005	68
NODE_448	Halomonas sp. 1513	Halomonas	Halomonadaceae	2.98	0.0001	0.0003	69
NODE_1461	Halomonas	Halomonas	Halomonadaceae	3.60	0.0001	0.0006	70
NODE_767	Halomonas sp. 1513	Halomonas	Halomonadaceae	2.52	0.0001	0.0006	71
NODE_109	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	3.15	0.0001	0.0003	72
NODE_332	Halomonas sp. 1513	Halomonas	Halomonadaceae	3.38	0.0001	0.0005	73
NODE_397	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	4.52	0.0001	0.0004	74
NODE_908	Halomonas beimenensis	Halomonas	Halomonadaceae	4.07	0.0000	0.0008	75
NODE_72	Halomonas aestuarii	Halomonas	Halomonadaceae	4.88	0.0000	0.0009	76
NODE_739	Halomonas sp. 1513	Halomonas	Halomonadaceae	3.68	0.0000	0.0008	77
NODE_43514	Bilateria	Spirometra	Diphyllobothriidae	0.99	0.0000	0.0006	78
NODE_411	Halomonas	Halomonas	Halomonadaceae	4.46	0.0000	0.0005	79
NODE_1601	Halomonas sp. 1513	Halomonas	Halomonadaceae	1.01	0.0000	0.0005	80

(table continues)

Table A.2 (Continued)

Contig	Annotation	Genus	familly	$-log_{10}(p\text{-}value)$	Imp2	Imp1	rank
NODE_146	Halomonas sp. 1513	Halomonas	Halomonadaceae	5.37	0.0000	0.0004	81
NODE_32405	Viruses	Pegivirus	Flaviviridae	1.10	0.0000	0.0004	82
NODE_1111	Halomonas aestuarii	Halomonas	Halomonadaceae	2.06	0.0000	0.0003	83
NODE_663	Halomonadaceae	Chromohalobacter	Halomonadaceae	1.93	0.0000	0.0003	84
NODE_27077	Homo sapiens	Homo	Hominidae	0.53	0.0000	0.0003	85
NODE_2264	Halomonas	Halomonas	Halomonadaceae	1.87	0.0000	0.0003	86
NODE_24450	Homo sapiens	Homo	Hominidae	1.36	0.0000	0.0003	87
NODE_480	Halomonas sp. 1513	Halomonas	Halomonadaceae	4.00	0.0000	0.0003	88
NODE_1114	Halomonas beimenensis	Halomonas	Halomonadaceae	1.63	0.0000	0.0003	89
NODE_2152	Gammaproteobacteria	NA	NA	4.82	-0.0000	0.0003	90
NODE_828	Halomonas sp. 1513	Halomonas	Halomonadaceae	4.78	-0.0000	0.0004	91
NODE_705	Halomonas	Halomonas	Halomonadaceae	4.00	-0.0000	0.0003	92
NODE_706	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	4.55	-0.0001	0.0003	93
NODE_2365	Halomonas sp. 1513	Halomonas	Halomonadaceae	2.92	-0.0001	0.0004	94
NODE_676	Halomonadaceae	Chromohalobacter	Halomonadaceae	4.41	-0.0002	0.0003	95
NODE_155	Halomonas	Halomonas	Halomonadaceae	5.07	-0.0002	0.0003	96

(table continues)

Table A.2 (Continued)

Contig	**Annotation**	**Genus**	**familly**	$-log_{10}(p\text{-}value)$	**Imp2**	**Imp1**	**rank**
NODE_12409	Viruses	Pegivirus	Flaviviridae	1.02	-0.0003	0.0003	97
NODE_438	Halomonadaceae	NA	Halomonadaceae	5.28	-0.0004	0.0008	98
NODE_409	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	3.96	-0.0005	0.0004	99
NODE_195	Bradyrhizobium sp. SK17	Bradyrhizobium	Bradyrhizobiaceae	4.48	-0.0005	0.0003	100