Humidor: Microbial Community Classification of the 16S Gene by Training CIGAR Strings with Convolutional Neural Networks

Douglas Sherman*

June 13, 2017

Abstract

We propose a new model for classifying microbiota based on CIGAR strings depicting indels and mismatches for 16S gene sequences. To test this model, we used the sequences found in the Green Genes database consisting of 23,662 aligned sequences and 237,880 unaligned sequences. We aligned the target sequences against consensus sequences generated by selecting the mode base pairs at each position for all aligned sequences under each of the 207 unique Orders. This resulted in 209,587 successfully aligned sequences against 180 of the 207 consensus sequences with on average 1322 alignments per reference. The resulting SAM files generated, on average, a 1433 dimensional feature set for each of the sequences with 39 non-null descriptors defining an insertion, deletion, or mismatch. These features defined the input for a Convolutional Neural Network following an architecture of 3 convolutional layers, a final deeply connected layer, and ReLu activation functions. The resulting model demonstrated similar accuracy to the RDP classifier with area of 0.719, 0.733, and 0.825 under the ROC curves for predicted Order, Family, and Genus respectively. Moreover, the model produced a likelihood for unclassifed sequences' taxonomic rank and was robust against incorrect alignments. We demonstrated that a new type of algorithm using CIGAR string information held up against the RDP classifier while offering a mechanism for determining the taxonomic rank of unknown sequences by taking advantage of the advances made in Deep Learning and Machine Vision.

*University of California - Davis, Mathematics and Computer Science Department

Contents

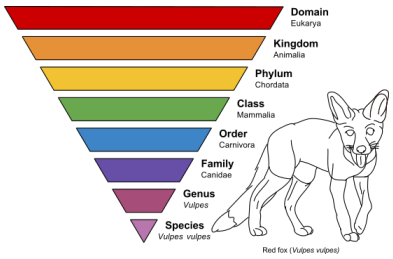

Figure 1: Taxonomic rank for all living beings. From top to bottom, the lower level is a subset of the higher.

1 Introduction

The microbes within the human gut have shown to play a substantial role in human psychological and physiological health, and also impact metabolic and immune functions. For example, intestinal inflammation is commonly believed to be associated with reduced bacterial diversity[1]. Moreover, excessive Fusobacterium has been linked to tumourigenesis [2] and irretable bowel syndrome [3], and there are some studies that show high levels of Firmicutes and low levels of Bacteroidetes have been found in mice with obesity [4]. Hence, the ability to indentify human gut microbiota has become increasingly important. Furthermore, algorithms that can efficiently and accurately classify this bacterial taxonomic heirarchy (See Figure 1) have become a focus in computational genetics. The 16S rRNA gene has become vital in this task as it is universally present across bacteria. Moreover, this gene can easily be amplified with universal primers, and is highly conserved except for nine hypervariable regions $V1 - V9$ [5]. An example of a classification algorithm that trains on the 16S rRNA hypervariable regions is the Ribosomal Database Project (RDP) classifier.

1.1 RDP Classifier

The RDP classifier is an algorithm that was developed using a Naive Bayes classifier approach to classifying bacterial rRNA sequences based on a large database [6]. The RDP classifier uses k-mers, or words, of 6 to 9 base pairs, and identifies these sub-sequences within N reference sequences following standard Natural Language Processing (NLP) mechanisms. The positions of these matches are ignored and only frequency of each word is considered. The Bayesian aspect of the algorithm comes from the word-specific and taxonomy-specific prior probabilities. For the word-specific priors, let $W = \{w_1, w_2, \cdots, w_d\}$ be the set of all k-mers and $n(w_i)$ be the number of sequences with w_i as an element, then we denote

$$P(w_i) = \frac{n(w_i) + 0.5}{N + 1}$$

as the prior probability for a specific word w_i. Moreover, for a genus G of M sequences and $m(w_i)$ of those sequences containing w_i, we have

$$P(w_i|G) = \frac{m(w_i) + P(w_i)}{M + 1}$$

Then the naivety of the algorithm comes from assuming independence between seperate words, which implies that the probability of a partial sequence $\{w_1, w_2, \cdots, w_f\} \subseteq S$ being in a given genus G is

$$P(S|G) = \prod_{w_i \in S} P(w_i|G)$$

The RDP classifier then impliments Baye's Rule to find the probability that this sequence is a member of the genus G as $P(G|S)$ [6]. To get more levels of a bacterial taxonomy, the RDP classifer would first classify the Kingdom, then traverse the taxonomic ranks down to the species level. This traversal also leads to compounded likelihood, or bootstrap confidence estimations, within each level of the taxonomy.

The Naive Bayes model used in the RDP classifier provides two limitations in its construction. First, it assumes that the probability of finding a word w_i in a genus G is indpendent of finding a word w_j in that same genus. This assumptions relies on the fact that the size of G is generally substantially larger than w_i, since if $|G| = |w_i|$ then it very clear that $w_i \in G$, implies $w_j \notin G$ provided $w_i \neq w_j$. Moreover, because this model traverses each level of the taxonomy in a single run, the likelihood of correctly predicting a given taxonomic rank must be less than or equal to the likelihood that it correctly predicted the parent rank.

1.2 OTU Clustering

As the database for bacterial sequences grew larger, the RDP and other 16S classifiers were unable to efficiently handle the substantial amount of data. Especially when some higher-level structure for the bacteria is unknown or is inconsistent[6], a solution to this problem is to first reduce the amount of the data before classifying using phylogenetic clustering. Phylogenetic clustering clusters sequences into Operational Taxonomic Units (OTUs) which define equivalence classes under some similarity metric with each class annotated by its representative's taxonomic rank.

OTU clustering algorithms measure the similarity between two sequences by aligning those sequences against each other and measuring the percent of matches. Historically, OTUs are defined by a 97% alignment match between species based on an empirical study in 2005 by Konstantinidis and Tiedje[7]. Once a set of representatives are determined, a seperate classification method is used to determine the taxonomy of each representative such as Naive Bayes[6] (Supervised) or Random Forest[8] (Unsupervised). There have been many concerns with OTU clustering algorithms, as the 3% similarity threshold ignores multiple substitutions occuring at the same location due to evolution[9], and it generally underestimates the number of substitutions when compared with an evolutionary metric such as multiple sequence aligment (MSA)[10]. For example, some the E.Coli genomes can range anywhere from 1 to 4.5 million base pairs; contradicting this 3% similarity threshold. However, the advancements in deep learning and big data allow us to classify large datasets without the need for dimensionality reduction techniques such as OTU clustering.

1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are many layered Artificial Neural Networks (ANN) that have revolutionized image classification and deep learning. Based on Fukushima's Neocognitron in 1980, CNNs allow for training of complex inputs by focusing on local connectivites between neurons, or nodes, of the network[11]. This drastically reduces complexity as each layer's nodes do not need to connect to every node of the previous layer as in a ANN. The CNN works by using a convolutional filter to break an image down into relatively much smaller overlapping regions, training the model on these regions, and then pooling them back together. This process often repeats multiple times with various transformations, or filters, being applied to each layer before training. Moreover, this tiling architecture of the model allows it to be translation invariant, as the weights of the CNN are trained on the individual regions rather than the image as a whole[12]. This invariance allows for important artifacts of the feature space to be recognized at any point within this space; such as a function group within a 16S gene sequence.

One of the key features of a CNN is the filters, or activation functions, used to transform the data at each layer of the network. Each filter processes the image so that the next layer can begin to recognize specific patterns within each image. Generally the first filter is a Convolutional filter that combines the neighboring pixels into a single pixel which not only reduced the dimensionality of the image but can also smooth out any noisy pixels that don't match their neighbors. Figure 2 shows an example of a 5×5 filter reducing a 32×32 image into a 28×28 image. Table 1 shows some filters that are commonly used in the other CNN layers.

Filter	Abbr.	Definition
Sigmoid Function	σ	$1/(1+e^{-x})$
Inverse Tangent	tanh	$\tanh(x) = 2\sigma(2x) - 1$
Softmax	$S_i(x)$	$\frac{exp(b_i+w_i x)}{\sum_j exp(b_j+w_j x)}$
Rectified Linear Unit	$ReLu$	$\max(0, x)$
Max Pooling	$Pool$	$a_{ij} = \max\{x_{\hat{i}\hat{j}} \vert \hat{i} \in [i-a, i+a], \hat{j} \in [j-a, j+a]\}$
Fully Connected	$Full$	A Fully Connected Layer as in a traditional ANN

Table 1: Example filters and activation functions for neural networks. The sigmoid, tanh, and Softmax activation functions are artifacts from ANNs, while Pooling, ReLU, and Fully Connected are filters often used in CNNs to refine an image into its underlying structure.

For traditional multi-layered neural networks, it was often found that the fully connected layers lead to high computational complexities and poor error rates[13]. CNNs were an exception to this rule, although its still relatively unclear as to why. In 1989, LeCun et al advanced CNNs using error gradients and demonstrated outstanding performace on handwriting recognition[14], setting the path for modern day CNNs to continue to be at the forefront of modern day deep learning. One hypothesis for CNNs success is that the localization of the neurons causes the error gradients to propagate without compounding error as they do with ANNs. Moreover, the addition of generative filters, such as rectified linear units and pooling, has led to outstanding results on the Caltech-101 dataset not just handwriting recognition [15].

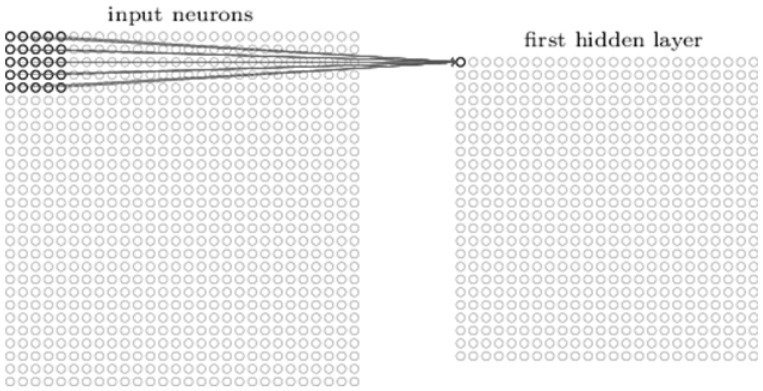

Figure 2: The first layer of a CNN is traditionally the convolutional layer which applies a convolutional filter to the input feature vector. In this visualization we see a 5×5 convolutional filter being applied to an input 32×32 image. The result is a $(32 - (5-1)) \times (32 - (5-1)) = 28 \times 28$ image as the last 4 columns/rows of the image do not have enough remaining entries to apply the filter. [16]

2 Models and Methods

Current alignment techniques rely on a known reference sequence for each species which is a luxury we do not often have with 16S gene microbiota. To avoid this we propose a method that aligns sequences against consensus sequences generated by consolidating the sequences of similar taxonomic rank. Then the alignment meta data, such as the CIGAR string that specifies how the target sequence aligned, can generate a thorough set of features for building a predictive model. Thus allowing us to classify microbes without any knowledge of their taxonomic rank *a priori*.

2.1 Consensus Sequence

In designing the Consensus Sequence, we had to consider developing a reference sequence that produced enough alignments to train on while ensuring these alignments had enough unique features that our model could recognize their differences. The consensus sequence were focues around the different branches of the taxonomic heirarchy for the 16S gene, as depicted in Figure 1. By consolidating all the sequences under a given Order, Family, or Genus we could guarantee that sequences within this class would only differ from the consensus sequence at the most relevant functional groups, thus generating CIGAR strings with mismatches displaying the most relevant information about each sequence.

We used the sequences available at the Green Genes 16s RNA Genes Database to compile our consensus sequences. This dataset consists of 23,662 sequences with full taxonomic rank, including 207 unique Orders, 354 unique Families, and 1067 unique Genuses. To construct the consensus sequence we selected all sequences of a given Order, Family, or Genus and built a synthetic sequence that consisted of the most common base among all the aligned sequences at each base. Figure 3 shows an example of a consensus sequence constructed from the three aligned sequences. We constructed a set of consensus sequences for all sequences in each Order, Family, and Genus which produced 207, 354, and 1067 unique consensus sequences respectively.

1:	.	.	A	C	A	_	C	C	.	.
2:	.	.	.	A	A	C	C	_	T	.
3:	.	.	.	C	T	G	_	C	.	.
C:	.	.	.	C	A	_	C	C	.	.

Figure 3: The consensus sequences were generated by selecting the mode base pair among all the aligned sequences of a given taxonomic Order, Family, or Genus. This figure illustrates 3 example sequences being consolidated into a single consensus sequence. Notice that the first and last base pair do not carry into the consensus sequence because the majority of the sequences had no base pairs at that position.

Green Genes also provided 237,880 unique unaligned sequences with either partial or full taxonomic rank defined. We then explored how well each of the unaligned sequences aligned to our consensus sequences. There were a few important details for determining which consensus sequences were going to be the best for our final model. First, we wanted to ensure that the highest percentage of target sequences would align to our consensus sequences. It was also important that there was minimal supplementary alignments, and that each alignment produced sequences near the expected 1500 base pair length after clipping. Moreover, we wanted a balanced spread of alignments so that we didn't end up with 90% of the target sequences aligning to one reference and the rest split among remaining references. Table 2 shows a summary of these characteristics for each of the groups of consensus sequences as well as the result of re-aligning the sequences after removing references that matched less than 200 and 2000 targets during the alignment process. The process we used to align the sequences is described in the following section. We concluded that the Order, untrimmed, concensus sequences for the model would produce the ideal results. This choice is primarily based on the CIGAR length as that dictates how many features we will have to build a predictive model off of; however, the number of aligned sequences was also an important factor as it gives a higher chance of target sequences not being thrown out.

2.2 Feature Set Generation

We used the a Burrows-Wheeler Alignment tool (BWA), based on the Burrows-Wheeler Transform (BWT), to align the target sequences against the consensus sequences. BWA "mimics the top-down traversal on a prefix trie generated from genome with a relatively small memory footprint"[17]. Specifically, we used the BWA-MEM algorithm since both the reference and target sequences were on the shorter side at around 1500 base pairs. We also compared with Bowtie aligner, but found no significant change in alignments or accuracy but saw a huge increase in computational time.

Rank	# Aligned	# Unique Ref	Avg. # per Ref	CIGAR Length
Order	209587	180	1321.56	26.77
Order 200	189855	74	3214.60	26.70
Order 2000	215487	43	5490.86	26.84
Family	43906	351	677.72	14.08
Family 200	49200	175	1359.3	14.31
Family 2000	86114	24	9776.04	16.27
Genus	46227	1030	230.95	12.15
Genus 200	56110	228	1043.34	13.17
Genus 2000	112202	22	10659.27	17.37

Table 2: This table depicts statistics for the aligments against specific consensus sequences. The **Rank** column lists on which taxonomic class we built the consensus sequences against; Order, Family, or Genus. It also defines if certain references were removed due to having less than 200 or 2000 matched sequences. **# Aligned** is the number of sequences that matched to a single reference sequence and **# Unique Ref** is the number of unique reference sequences that aligned to at least one target sequence. The **Avg. # per Ref** column is the average number of non-supplementary sequenes that aligned to each reference sequence and **CIGAR Length** is the average number of non-numeric characters in the CIGAR strings (i.e 3M2D would have lenth 2). We clearly see that removing referene sequences causes the number of unique references to decrease and number of alignments per reference to increase, but this increases the length of the CIGAR string. Likely because it causes the sequences that would normally align to the removed references to force an alignment against a less ideal reference; increasing the number of indels or mismatches.

Once we aligned the target sequences against the consensus reference sequences, we obtained a set of .sam files. These .sam files provided information about each alignment including CIGAR and MD strings. The CIGAR strings map out how each base pair of the target sequence aligns against the most similar reference sequence, and the MD string provided insert and deletion information as described in Table 3. Our model is built on the idea that theses CIGAR and MD strings contain enough information to accurately predict the target sequence's taxonomy, and the featureset used in our predictive model is generated entirely from these fields combined with the position (POS) tag.

To convert this information into a featureset we had to parse the CIGAR string into something a model could train. Since we designed the reference sequences in a way that only the dissimilar base pairs should be informative, we focused on labeling the Inserts, Deletes, and Substitutions against the consensus sequence. To convert this information into informative features, we defined a featureset the length of the consensus sequence and then each of these features would be valued one of D if there was a deletion from the reference sequence, A, C, T, or G if one of these base pairs was inserted from the target sequence, or an X if a region presented a mismatch. Any other event, such as a match, soft-clip, or hard-clip was simply left as a NULL value for that feature. An example of a feature set generated from the alignment of a target sequence to a given reference sequence is outlined in Figure 4.

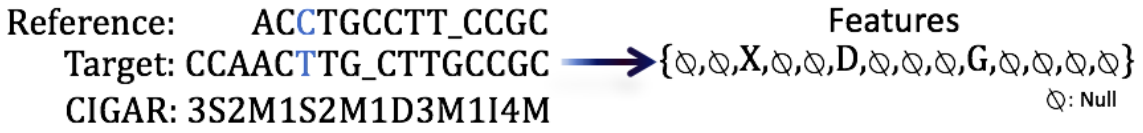

Figure 4: The process of converting from a CIGAR string to a feature set for a given target sequence/reference sequence pair. Notice that soft-clippings and matches are ignored as they do not provide any unique information for each sequence. Inserts are also labeled as the specific SNP that is inserted as for many variants a single SNP or MNP dictates a functional group. Moreover, this causes most of the features to be null values, which could needlessly increase the complexity of many machine learning models.

Op	Description
M	alignment match (can be a sequence match or mismatch)
I	insertion to the reference
D	deletion from the reference
N	skipped region from the reference
S	soft clipping (clipped sequences present in target sequence)
H	hard clipping (clipped sequences not present in target sequence)
P	padding (silend deletion from padded reference)
=	sequence match
X	sequence mismatch

Table 3: A Table Depicting Sam File Informaiton [18]

The average consensus sequence was 1443 base pairs, and by aligning against these synthetic reference sequenes we obtained on average of only 39 indels or mismatches for each target sequence. This left our featureset consisting of nearly all NULL values. This allowed for us to train the model on only 39 characteristics so long as we can handle the 1400 NULL values in a way that does not significantly add to the complexity of the model.

2.3 Convolutional Neural Network

Our decision to use a Convolutional Neural Network (CNN) to classify the microbes was due to the nature of the generated feature set. Because the each sample had mostly NULL features, we needed a model that could would pick out only the relevent characteristics of each extended CIGAR string and not increase significantly by the NULL features. CNNs are traditonally used for image recognition and are able to discern the relevant pieces of an image and then combine these pieces in order to classify that image. This is achieved by subsequent convolutions and max pooling layers to first consolidate neighboring pixels and then select the maximal pixel value in this consolidation. Using this layering mechanism we can select out only the relevant funcitonal groups. Then the sum total of these functional groups provides a likelihood of being a certain microbe, just as an image requires a nose, two eyes, and a mouth to be considered a face. Moreover, since aligning to a consensus sequence simply picked out the unique sections for each microbe, this allowed our CNN to solely focus on the relevant groups that uniquely differentiate each microbe. Figure 5 depicts the parsed CIGAR strings as images after hashing any strings and assigning numeric values to D and X. In this image we see that these images become more similar as we traverse the taxonomic heirarchy, lending support to the effectiveness of a CNN for prediction. For our CNN chose an architecture given by

$$Conv_{8\times 8} \implies ReLu \implies Pool \implies Conv_{32\times 32} \implies ReLu \implies Pool \implies Full \implies ReLu \implies Out$$

Using this architecture allowed us to the divide the CIGAR string into the seperate functional groups that dictate which species it refers to then use a regular Neural Network to classify on only those functional groups.

3 Results and Analysis

This model hinged on three main assumptions. First, aligning against the consensus sequences would group the target sequences into the most similar sequences. Second, parsing the CIGAR string would produce enough features to classify the sequences without having too many redundant features shared between all the sequences in a given group. Finally, the Convolutional Neural Network could take advantage of the sparsity of the features for effeciency and maintain high prediction accuracy based solely on the main functional groups of each sequence.

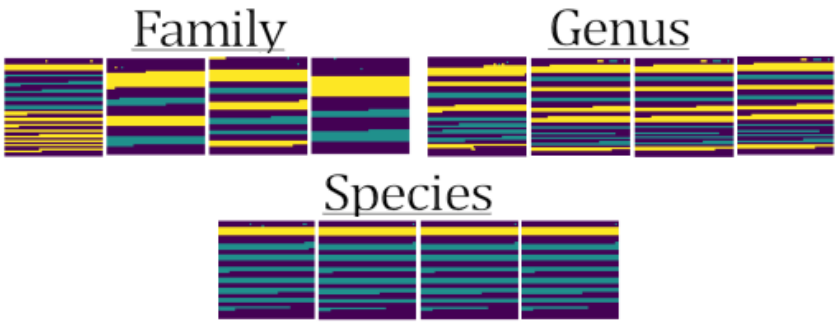

Figure 5: These are the image representation of the featuresets generated for a specific Family, Genus, and Species. The family depicted is *PRR-10* and we see that each of the 4 images is significantly different. The genus is *Planococcus* and the images begin to converge to a similar view. The species is *Rhizobiummesosinicum* and the images are nearly exactly the same. The convergence of the images show that by the Species level, the relevant indels and mismatches should be indentical allowing for the CNN to train on.

3.1 Consensus Sequence Alignment

How the target sequences aligned against the consensus sequences is displayed in Figure 7 at the end of the paper. Each symbol represents a target sequence, and the color & shape of the symbol depicts the first point along the intended reference sequence's taxonomy that the target and reference match. Only Bacteria are shown in this tree, so sequences that don't match until the Kingdom level are considered unmatched. A pattern that arose is that the reference sequences with the most target alignments for a given order consensus sequence were the least likely to match order to order. Figure 8 shows just the sequences that did not match at the order level. A possible remedy for this is to generate consensus sequences at the Family or Genus level for orders that had many more matches than some threshold number. However, the accuracy did not show any trend when plotted against the number of target sequences that aligned for each reference sequence. Thus, when the sequences aligned against the wrong order, it did not significantly impact the effectiveness of the model.

3.2 Comparison with RDP

This algorithm was designed to be an alternative for pre-existing classification algorithms such as the RDP classifier. We compared the accuracy of our model to the RDP using the 16S green genes data base sequences. The ROC curves at Figure 6 show the accuracy of the Humidor when trying to predict Order, Family, and Genus for the target sequences, and the RDP classifier's resulting accuracy. These ROC curves show Sensitivity vs. 1-Specificity where Sensitivity is the True Positive Rate (TPR) and Specificity (SPC) is the True Negative Rate given by

$$TPR := \frac{TP}{P} = \frac{TP}{TP + FN}$$
$$SPC := \frac{TN}{N} = \frac{TN}{TN + FP}$$

where TN and TP are the negative and positive samples correctly identified as negative or positive respectively, and FP and FN are Type 1 and Type 2 error rates. Note that positive and negative samples indicate a binary classification, so these statistics are computed based on an average one vs. all method where TP is

exactly computed as

$$TP := \frac{1}{|C|} \sum_{c \in C} \frac{1}{|c|} \sum_{x_i \in c} \begin{cases} 1 & \text{if} \quad x_i = \hat{x_i} = c \\ 0 & \text{else} \end{cases}$$

where $\hat{x_i}$ is the predicted value and c is one of the values of all the classes C for some Taxanomic rank. These ROC curves show that both the Humidor and RDP classifers produce similar results when trying to classify on Genus, Order, or Family Ranks.

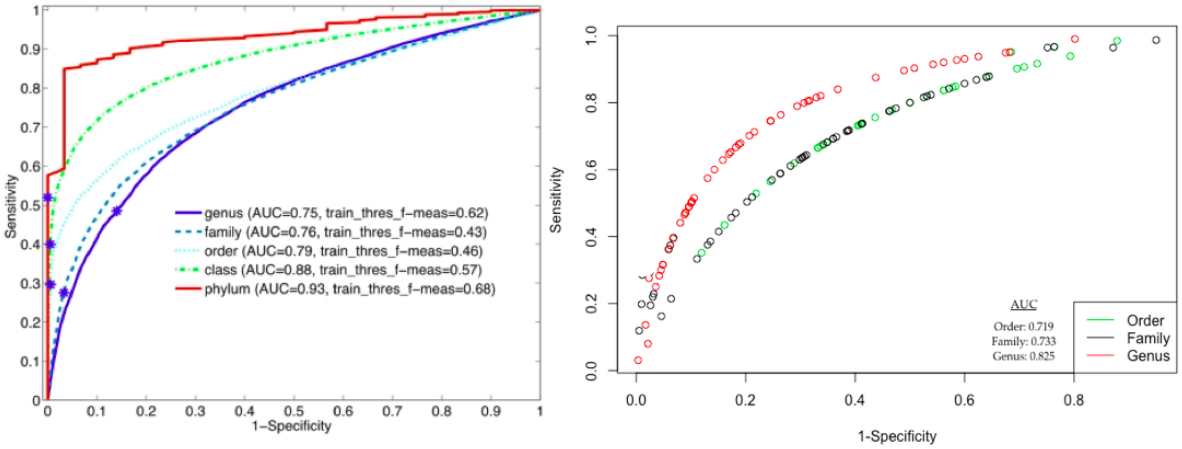

Figure 6: This shows ROC curves for both the Humidor and RDP classifiers[20]. We see that for Order, Family, and Genus we have similar area under the curves. For the RDP model, the taxonomic heirarchy must be followed in order, so error in the higher ranks is propagated to the lower ranks. Thus from Phylum to Genus, the accuracy is decreasing. For the Humidor model, each rank under the consensus rank is classified seperately and thus this restriction does not apply as shown in the right ROC curve.

3.3 Conclusion

The Humidor classifer uses the advances of image detection to create an alignment based classifcation algorithm that trains on CIGAR strings. By initially implimenting the Burrows-Wheeler alignment algorithm against consensus sequences, the target sequences are first divided by their most probable Order, then we obtain CIGAR strings with only the most informative variants for each Family, Genus, or Species. The ROC curves showed that Humidor performs similarly to the RDP classifier and the CNN model has been tuned to handle huge datasets without the need of clustering to reduce dimensionality. Moreover, in the event of being unable to classify the Species of a given sequence, the likelihood measures could be traced up the heirarchy to determine the most likely Genus, Family, or Order instead.

Future work would warrant refining a procedure for handling uncertain or unclassified predictions. The OTU clustering algorithms suffer from having to generalize every possible Species. However, by first minimizing the number of samples that have to be clustered with Humidor, a more data-oriented approach can be made with the clustering algorithm. For example, using spectral clustering or diffusion maps to cluster on the underlying structure of the data since most of the biological variances would be handled by the Humidor classifier. Furthermore, as the study of CNNs develops, more effort can be placed in providing a specialized CNN architecture.

References

[1] Guinane, C. M., & Cotter, P. D. (2013). Role of the gut microbiota in health and chronic gastrointestinal disease: understanding a hidden metabolic organ. *Therapeutic Advances in Gastroenterology*, 6(4), 295-308. http://doi.org/10.1177/1756283X13482996

[2] Kostic A., Gevers D., Pedamallu C., Michaud M., Duke F., Earl A., et al. (2012) Genomic analysis identifies association of Fusobacterium with colorectal carcinoma. Genome Res 22: 292-298 PubMed: 22179717

[3] Strauss J., Kaplan G., Beck P., Rioux K., Panaccione R., Devinney R., et al. (2011) Invasive potential of gut mucosa-derived Fusobacterium nucleatum positively correlates with IBD status of the host. Inflamm Bowel Dis 17: 1971-1978 PubMed: 21830275

[4] Ley R., Backhed F., Turnbaugh P., Lozupone C., Knight R., Gordon J. (2005) Obesity alters gut microbial ecology. Proc Natl Acad Sci U S A 102: 11070-11075 PubMed: 16033867

[5] Nguyen, N., Warnow, T., Pop, M., and White, B. (2016) A perspective on 16S rRNA operational taxonomic unit clustering using sequence similarity. *Biofilms and Microbiomes* 16004(2) doi:10.1038/npjbiofilms.2016.4

[6] Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007). NaÃ¯ve Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy. *Applied and Environmental Microbiology*, 73(16), 5261-5267. `http://doi.org/10.1128/AEM.00062-07`

[7] Konstantinidis, K. T. and Tiedje, J. M. Genomic insights that advance the species definition for prokaryotes. (2005) *Proc. Natl Acad. Sci.* 102, pp. 2567-2572.

[8] Chaudhary, N., Sharma, A. K., Agarwal, P., Gupta, A. and Sharma, V. K. 16S classifier: a tool for fast and accurate taxonomic classification of 16S rRNA hypervariable regions in metagenomic datasets. (2015) *PLoS ONE* 10, e0116106.

[9] Yang, Z. Computational Molecular Evolution. Oxford Univ. Press, (2006).

[10] Rosenberg, M. S. Evolutionary distance estimation and fidelity of pair wise sequence alignment. (2005) *BMC Bioinformatics* 6(102).

[11] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position (1980) *Biological Cybernetics*, 36, pp. 193-202.

[12] Korekado, K; Morie, T; Nomura, O; Ando, H; Nakano, T; Matsugu, M; Iwata, A (2003). A Convolutional Neural Network VLSI for Image Recognition Using Merged/Mixed Analog-Digital Architecture. Knowledge-Based Intelligent Information and Engineering Systems: 169-176.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. (2007) *Advances in Neural Information Processing Systems MIT Press* 19, pp. 153-160.

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hub- bard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. (1989) *Neural Computation* 1(4), pp. 541-551.

[15] Bengio, Y. Foundations and Trends in Machine Learning (2009). (2)1 pp. 1-127 DOI: 10.1561/2200000006

[16] Deshpande A. (2016). A Beginner's Guide To Understanding Convolutional Neural Networks. A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

[17] Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, 25:1754-60. PMID: 19451168

[18] https://samtools.github.io/hts-specs/SAMv1.pdf

[19] Lan, Y., Wang, Q., Cole, J. R., & Rosen, G. L. (2012). Using the RDP Classifier to Predict Taxonomic Novelty and Reduce the Search Space for Finding Novel Organisms. *PLoS ONE*, 7(3), e32491. http://doi.org/10.1371/journal.pone.0032491

[20] DeSantis, T. Z., P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen. 2006. Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. *Appl Environ Microbiol* 72:5069-72.

[21] DeSantis, T. Z., I. Dubosarskiy, S. R. Murray, and G. L. Andersen. 2003. Comprehensive aligned sequence construction for automated design of effective probes (CASCADE-P) using 16S rDNA. *Bioinformatics* 19:1461-8.

[22] Chen W, Zhang CK, Cheng Y, Zhang S, Zhao H (2013) A Comparison of Methods for Clustering 16S rRNA Sequences into OTUs. *PLoS ONE* 8(8): e70837. https://doi.org/10.1371/journal.pone.0070837

[23] Schloss, P. D., Westcott, S. L., Ryabin, T., Hall, J. R., Hartmann, M., Hollister, E. B., Lesniewski, R. A., Oakley, B. B., Parks, D. H., Robinson, C. J., Sahl, J. W., Stres, B., Thallinger, G. G., Van Horn, D. G., and Weber, C. F. (2009) Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities. *Appl Environ Microbiol*, 75(23) http://aem.asm.org/content/75/23/7537.short

[24] Department of Microbiology and Immunology - University of Michigan, Mothur. https://mothur.org/

[25] Stackebrandt, E., W. Frederiksen, G. M. Garrity, P. A. D. Grimont, P. Kampfer, M. C. J. Maiden, X. Nesme, R. Rossello-Mora, J. Swings, H. G. Truper, L. Vauterin, A. C. Ward, and W. B. Whitman. 2002. Report of the ad hoc committee for the re-evaluation of the species definition in bacteriology. *Int. J. Syst. Evol. Microbiol.* 52:1043-1047. PubMed: 12054223

[26] Bishop, C. M. Pattern Recognition and Machine Learning (2006). Springer, ISBN-10: 0-387-3107308

[27] Burrows M, Wheeler DJ. Technical report 124. Palo Alto, CA: Digital Equipment Corporation; 1994. A block-sorting lossless data compression algorithm.

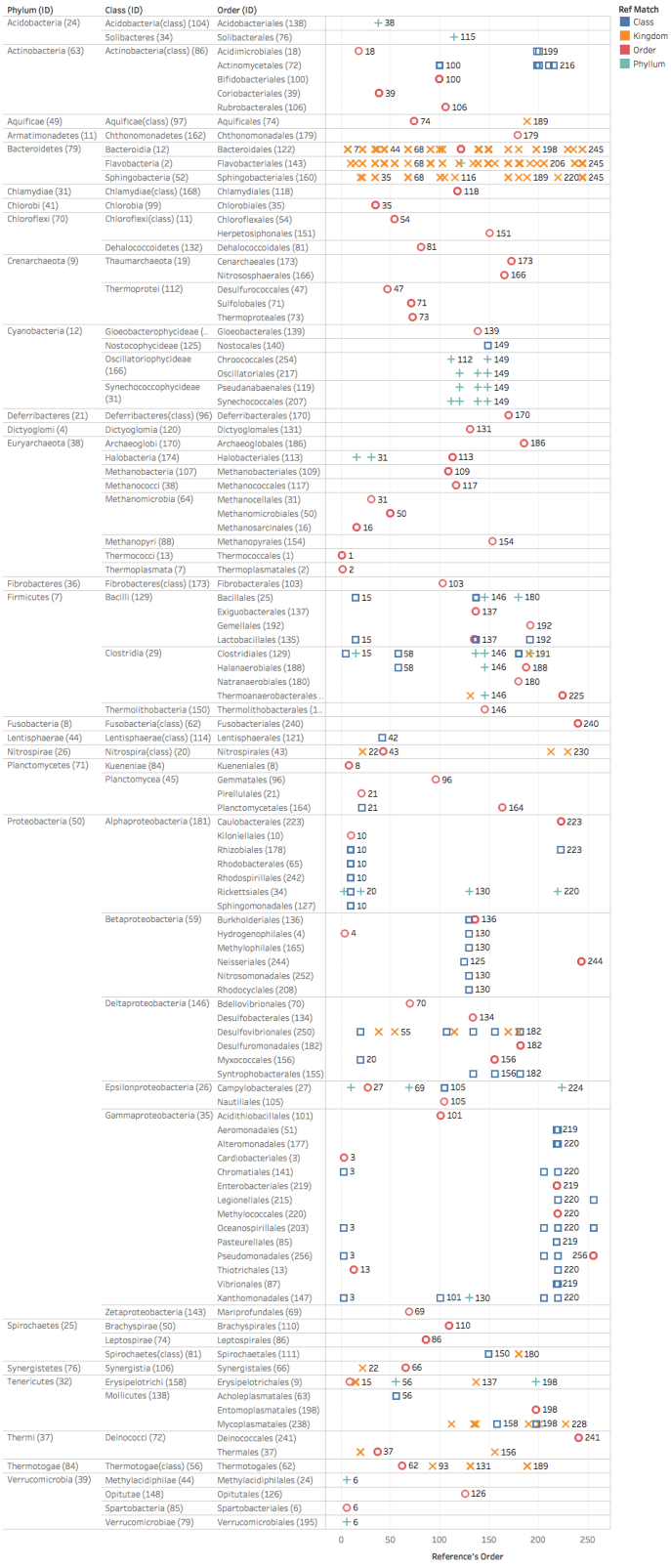

Figure 7: A heirarchy tree depicting the aligned sequences to each reference sequence.

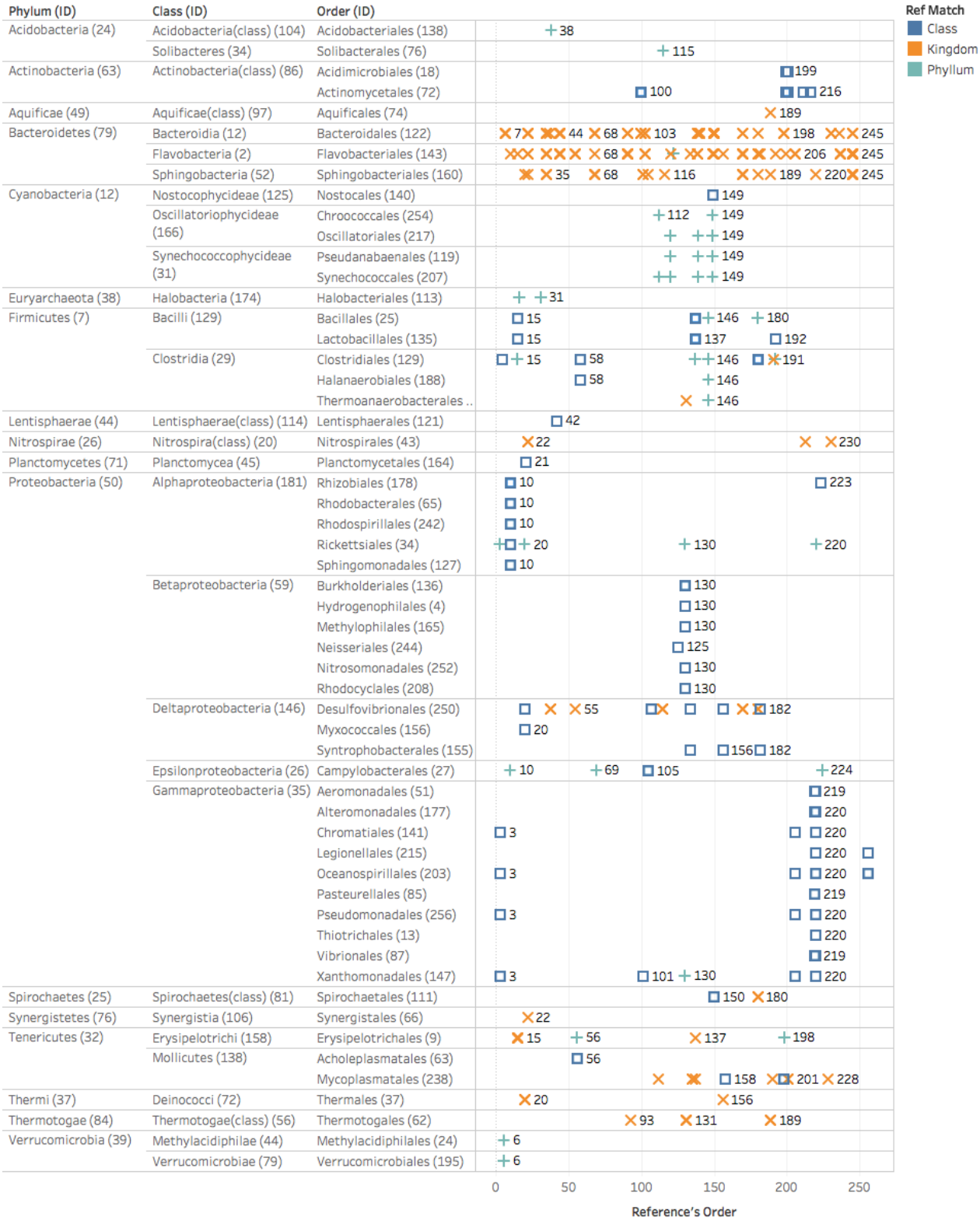

Figure 8: Another heirarchy depicting the aligned sequences, but the correctly aligned sequences are removed. We see that reference sequence with many alignments had the highest number of incorrect alignments.