Kathleen Hablutzel
BIO 334/BIO 335
17 December 2021

# Classifying Viral Host from Codon Usage Bias

## Summary

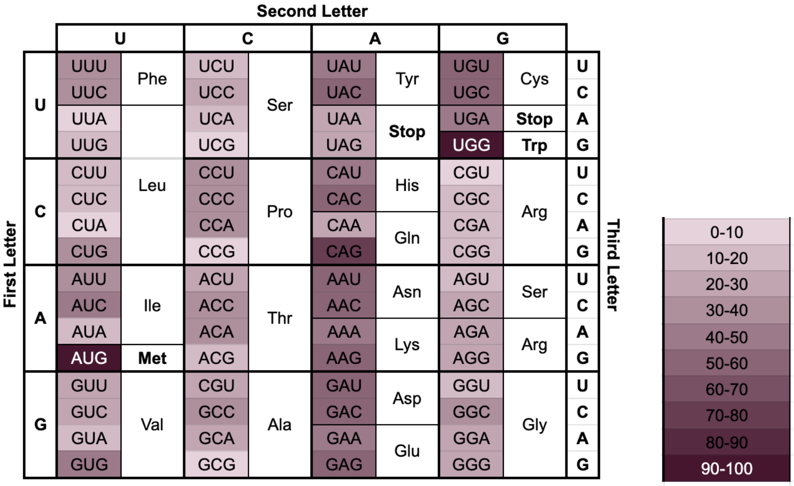
This project compiles a database of the codon usage bias of thousands of animals, plants, and viruses from the NCBI RefSeq database (https://ftp.ncbi.nlm.nih.gov/genomes/refseq/). We build a classifier of whether a virus infects humans or not. Despite a small sample size, we achieve a 75% accuracy at predicting whether a virus infects humans based on codon usage distribution alone. We also test building a classifier from only biased regions of the viral genome, with no improvement to classifier accuracy. However, the success of these classifiers suggests that codon usage bias may be a useful predictor of whether a virus may infect humans, and could be potentially incorporated as a factor in models that predict emerging viral threats to the human species. Other potential applications of this project could recycle the NCBI RefSeq genome scraper to build datasets of any statistic about every RefSeq genome (or a subset thereof). Additionally, the codon usage dataset from this work could be distributed for the convenience of bioinformaticians and machine learners alike, as no known dataset of codon usage currently exists.

## Introduction

### Codon Usage Bias (CUB)

Most organisms show disproportionate usage of synonymous codons for a given amino acid. For each of the 18 degenerately coded amino acids—those with multiple synonymous codons available in the universal codon chart—some codons appear more frequently and thus have more available tRNAs for translation. For example, in the human genome, three of the four proline codons (CCU, CCC, and CCA) are used about 30% of the time each, while the fourth codon (CCG) only appears about 10% of the time a gene calls for the proline amino acid. Meanwhile, other degenerate amino acids show more proportional codon usage, such as in the duet amino acid asparagine in the human genome. The corresponding AAU and AAC codons are used in almost exactly 50/50 proportions over the entire human genome (Figure 1).

In 1990, Wright developed an effective codon usage statistic ($N_C$) to measure the extent of the codon bias of a genome. Traditionally, $N_C$ ranges between 20 and 61, where 20 represents a genome using only one codon for each amino acid, while 61 represents a genome making proportionate use of every available codon. (Only 61 codons are available for amino acids, as three are used for stop codons.) In humans, $N_C$ is approximately 55, indicating that the human genome's disproportionate codon usage creates an imbalance equal to missing six entire codons from the genome. Banerjee et al. (2005) group effective codon usages into three ranges, with the moderate bias group starting at an effective codon usage of 50 or below. In this work, we use this figure as the threshold for whether a region of a genome is biased or not.



|  | Second Letter |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| First Letter | U |  | C |  | A |  | G |  | Third Letter |
| U | UUU | Phe | UCU | Ser | UAU | Tyr | UGU | Cys | U |
|  | UUC |  | UCC |  | UAC |  | UGC |  | C |
|  | UUA |  | UCA |  | UAA | Stop | UGA | Stop | A |
|  | UUG |  | UCG |  | UAG |  | UGG | Trp | G |
| C | CUU | Leu | CCU | Pro | CAU | His | CGU | Arg | U |
|  | CUC |  | CCC |  | CAC |  | CGC |  | C |
|  | CUA |  | CCA |  | CAA | Gln | CGA |  | A |
|  | CUG |  | CCG |  | CAG |  | CGG |  | G |
| A | AUU | Ile | ACU | Thr | AAU | Asn | AGU | Ser | U |
|  | AUC |  | ACC |  | AAC |  | AGC |  | C |
|  | AUA |  | ACA |  | AAA | Lys | AGA | Arg | A |
|  | AUG | Met | ACG |  | AAG |  | AGG |  | G |
| G | GUU | Val | GCU | Ala | GAU | Asp | GGU | Gly | U |
|  | GUC |  | GCC |  | GAC |  | GGC |  | C |
|  | GUA |  | GCA |  | GAA | Glu | GGA |  | A |
|  | GUG |  | GCG |  | GAG |  | GGG |  | G |

Figure 1: The codon usage distribution of the human genome. Darker shading on a codon indicates that it is used more frequently among all its amino acid's codons.

As another way to quantify codon usage, Bahir et al. (2009) represent the codon usage of an organism as a vector of 59 proportions—one for each amino acid-coding codon except those for methionine and tryptophan. These two amino acids are commonly excluded from codon usage bias analysis, since a single codon corresponds with each of these amino acids, and is thus used 100% of the time for the amino acid. For the other eighteen amino acids, either two, three, four, or six of the 59 remaining codons correspond with the amino acid. The entry in the codon usage vector at each codon is the proportion of references of this codon to references of any synonymous codon for its amino acid. From these vectors of 59 codon usage proportions, differences in codon usage between two organisms can be calculated as a distance between vectors such as an L1 norm (Manhattan distance) or L2 norm (Euclidean distance).

### Codon Usage Bias in Viruses

Viruses have been observed to favor synonymous mutations which bring their codon usage distribution closer to that of their host organism (Bahir et al., 2005; Chen et al., 2020; Hershberg & Petrov, 2008;  Jenkins & Holmes, 2003; Mollentze et al., 2021). Since viruses rely on host cell tRNAs to build viral proteins, matching a host's codon usage helps maximize translational efficiency. However, to avoid competition with the host cell's own proteins, viral genomes have also been observed to diverge in codon usage when their distribution is too similar to that of their host. Chen et al. (2020) observe dissimilation between virus and host codon usage biases when CUB grows too similar, as the viral translation process starts impeding host translation.

In human viruses, Bahir et al. (2009) observe strong resemblances in codon preference to the human host. This pattern does not hold for viruses which infect other mammals or aves. However, since Bahir et al.'s work finds human-infecting viruses to have similar codon usage distributions to humans, these viruses may also have similar codon distributions with each other. We wonder if potential codon usage similarities among human-infecting viruses may be enough to distinguish human viruses from other viruses. Although Gong et al. (2020) find that codon bias analysis may be insufficient for discovering hosts of SARS-CoV-2, other work has successfully classified viruses as threats to the human species. Mollentze et al. (2021) train a classifier to predict the probability that viruses will be able to infect humans from the viral genomic sequences. Thus, this work attempts to build a similar classifier to predict whether a virus infects humans from its codon usage bias alone.

### Research Aims

In this paper, we aim to build a model which classifies viruses with a human or non-human host based solely on their genome's codon usage distribution. To that end, we build a dataset of codon usage for 9701 genomes and train a machine learning classifier to predict whether a virus infects humans. We observe that we may distinguish viral hosts with moderate accuracy based on codon usage distribution alone. Working from full viral genomes or only the more biased regions of the genome does not significantly alter the accuracy of this classifier (75% for full genomes vs. 70% for only biased genomes). However, the success of this classifier implies that some aspect of the codon usage of human viruses is distinct from other viruses, suggesting potential to detect new viral threats to the human species from codon usage distributions in addition to other factors.

## Materials and Methods

### Genome Collection

Coding sequences for all organisms were collected from the NCBI RefSeq Genomes project database (https://ftp.ncbi.nlm.nih.gov/genomes/refseq/). This database is a subset of NCBI's GenBank database, including only highly annotated submissions which have been selected as reference genomes for a species. We utilized the reference assembly for eukaryotic genomes, but since no official reference assembly is identified for viral genomes, we instead utilized the latest genome assembly.  For any given genome assembly, we downloaded only the coding domains (CDS) of the genome, accessible at https://ftp.ncbi.nlm.nih.gov/genomes/refseq/{taxon}/{species}/{assembly_group} /{assembly}/{assembly_cds_from_genomic.fna.gz}, where:

- **taxon** is the RefSeq taxonomic group for the genome, such as "vertebrate_mammalian", "vertebrate_other", "invertebrate", "plant", or "viral",

- **species** is the species binomial for eukaryotes and prokaryotes, or the assigned viral name for viruses,
- **assembly_group** is the type of assembly—either "reference" for eukaryotes and prokaryotes, or "latest_assembly_versions" for viruses,
- **assembly** is the assembly information for the genome, in the format [assemblyAccession.version]_[assemblyName],
- and **cds_from_genomic** is the desired genome format, among other options such as "genomic", "protein", and "rna".

For example, the CDS for the human reference genome is available at https://ftp.ncbi.nlm.nih.gov/genomes/refseq/vertebrate_mammalian/Homo_sapiens/reference/GCF_000001405.39_GRCh38.p13/ GCF_000001405.39_GRCh38.p13_cds_from_genomic.fna.gz. Each genome was accessed in an automated fashion using a custom-built Python script, available on GitHub in the project repository (https://github.com/krhablutzel/codon-usage-bias/blob/main/generate_data.py). See **Appendix 1** for more information on this NCBI RefSeq database web scraper.

We chose to work solely from the coding domains of each genome to better guarantee that we read each codon in the correct reading frame. The NCBI RefSeq database trims CDS features which begin in frame two or three by one or two base pairs such that every CDS gene begins with the first complete codon. This guarantees that we may read codons in the correct reading frame; however, as a tradeoff, we limit our analysis to the exons of each genome. We accept this limitation in our data collection, as nucleotides are only read together as codons in the context of translation.

### Codon Usage Calculations

Once we obtain each genome, we calculate codon usage frequencies within each degenerately coded amino acid. For each of these 18 amino acids, the frequencies of its corresponding codons sum to 1. Since methionine and tryptophan each utilize only one codon, we excluded these amino acids from our analysis. We also excluded stop codons, which were not included in most CDS sequences. Thus, each of the 59 redundant amino acid-coding codons was assigned a number between 0 and 1. In addition to codon usage frequencies, we also calculate Wright's effective codon usage statistic ($N_c$) for each genome and record metadata such as sequence length and assembly information for later reference. Then, since codon usage bias varies among genes, and regions of low bias may drown out the signal of more strongly biased regions, we repeat our codon usage calculations for the genome using only genes with at least moderate bias (as measured by an effective codon usage below 50—a threshold we obtained from the work of Banerjee et al. (2005)). As a result, our dataset contains two codon usage calculations for each genome—one for the full genome, and one for only the more biased regions of the genome. For more information on our dataset creation process, see **Appendix 1**.

## Viral Host Collection

For viral genomes, we collected additional viral host information from the “Accession list of all viral genomes,” available from the NCBI Viral Genomes resource page (https://www.ncbi.nlm.nih.gov/genome/viruses/). If “human” is listed in the host column of this dataset, we count a virus as human-infecting, even if the virus has other potential hosts as well. For example, Cowpox virus infects human and vertebrate hosts and thus is labeled as a human virus. All other viruses are labeled as non-human. We join this viral host dataset with the CUB dataset on species name to match viruses with their host labels. Since the viruses contained in the RefSeq database and the Accession list are not identical sets, our analysis is limited to viruses contained in both datasets when we work with host labels of viral genomes. This set intersection consists of 4660 viruses.

## Initial Visualization

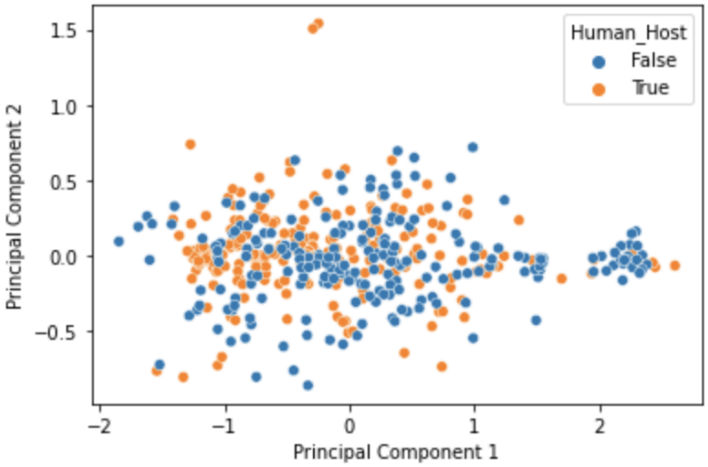


Figure 2: Viral codon usage summarized in two dimensions. We filtered the data for an equal number of human and non-human viruses such that the non-human viruses would not overwhelm the plot.

To obtain an initial picture of how codon usage varies among viral genomes, we utilized principal component analysis to summarize codon usage variation in two dimensions (Figure 2). Differences in codon usage were not immediately apparent between human and non-human viruses. However, we lose a great deal of variation when we reduce our data to two dimensions rather than the original 59 (one dimension for each degenerate codon). As a result, a pattern to distinguish human and non-human viruses may be more easily discovered when working with the full 59 dimensions of variation, so we decided to continue with our analysis.

## Preparing the Dataset

Our dataset contained significantly more non-human viruses than human viruses (4440 vs. 220 non-human). This could be potentially problematic in training our model, as a model which guesses non-human 100% of the time would have an accuracy of 95%. Since an incredibly imbalanced dataset could result in a less useful model, we filtered our dataset down to the 220 human viruses and 220 randomly selected non-human viruses, for a total of 440 viruses in our dataset. Then, we randomly divided 90% of our data into a test-train dataset (396 viruses) and reserved the other 10% for model validation (44 viruses). All code behind this analysis is available in the supplementary materials on GitHub in the analysis_full_genome.pdf and helper.py files**.** All data manipulation is through Python's Pandas package, data visualization is through the Matplotlib and Seaborn packages, and models are generated with the Scikit-learn package.

**Model Selection and Training**

To select which classifier may best predict viral hosts, we performed 10-fold cross-validation on the test-train dataset. In 10-fold cross validation, the test-train dataset is divided into ten randomized equally-sized "folds" of observations. For each classifier, we build a model with that classifier 10 times, each time training the model on nine of these folds and testing the model's accuracy on the remaining tenth fold. Then, we average errors across the ten models to obtain the cross-validation error for that classifier type. For this dataset, the classifier attempts to predict whether a virus infects humans (True or False) from the virus's codon usages of the 59 degenerate codons. Error represents the proportion of viruses the model labeled incorrectly (either labeling a human-infecting virus as non-human, also known as a False Negative, or labeling a non-human-infecting virus as human, also known as a False Positive).

| Classifier | Cross-Validation Error |
|---|---|
| Decision Tree | 26.3% |
| k-Nearest Neighbor | 29.8% |
| Random Forest | 31.1% |
| Support Vector Machine | 36.9% |

Table 1: Cross-validation errors for the viral host classifier.

We ran our cross-validation with four kinds of classifier: Decision Tree, Random Forest, k-Nearest Neighbor (kNN), and Support Vector Machine (SVM). Of these classifiers, the Decision Tree Classifier exhibited the lowest cross-validation error and was selected as the best model (Table 1). Accordingly, we trained a Decision Tree Classifier on the entire test-train dataset to obtain our model for predicting viral hosts.

**Model Selection and Training for Biased Regions of the Genome**

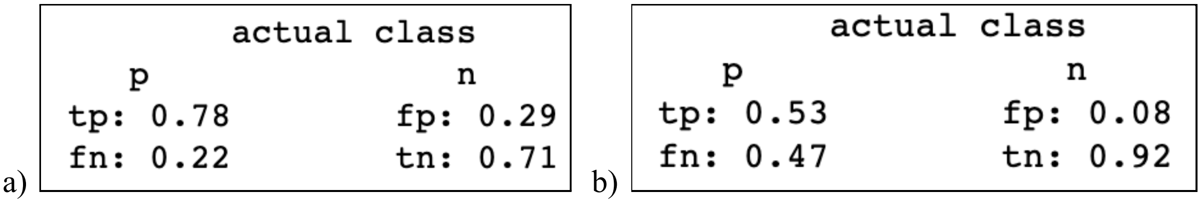
We repeated the process of model selection (via 10-fold cross-validation) and training on the dataset which calculates codon usage bias from only biased regions of the genome. Some viruses did not have enough biased regions to accurately calculate codon usage bias, so this dataset included only 308 viruses (154 human and 154 random non-human viruses). As before, we reserved 10% of the data (30 viruses) for model validation and performed model selection and trained on the remaining 90%. After 10-fold cross-validation, we selected a Decision Tree Classifier as the best model of the data and trained the model on the test-train dataset (Table 2).

| **Classifier** | **Cross-Validation Error** |
|---|---|
| Decision Tree | 24.4% |
| Random Forest | 27.4% |
| k-Nearest Neighbor | 27.4% |
| Support Vector Machine | 40.7% |

Table 2: Cross-validation errors for the viral host classifier built from only biased regions of the viral genomes.

## Results

For both the full-genome and biased-region classifiers, we validated the model's performance on the validation set, consisting of viruses never-before-seen by the model. The full-genome classifier achieved 75% accuracy, and the biased-region classifier achieved 70%. The full-genome classifier achieved high rates of both true positives and true negatives, while the biased-region classifier achieved a high true negative rate and a fairly low true positive rate (see Figure 3 for the confusion matrices).



a)

```
        actual class
   p                n
tp: 0.78        fp: 0.29
fn: 0.22        tn: 0.71
```

b)

```
        actual class
   p                n
tp: 0.53        fp: 0.08
fn: 0.47        tn: 0.92
```

Figure 3: a) The confusion matrix for the full-genome classifier. b) The confusion matrix for the biased-region classifier.

## Discussion

### Codon Usage Bias Alone is a Moderately Useful Predictor of Viral Host

The moderate accuracy of the full-genome model indicates that the model is learning features of codon usage variation which distinguish human-infecting viruses from other viruses. If our model was guessing at random, we would expect accuracies near 50%. Additionally, our model is not simply applying one label to every piece of data—with high true positive and true negative rates, our model is accurate at deciding when a virus does infect humans as well as when it does not. Among our 59 dimensions of variation, the model has found a pattern to distinguish human viruses moderately accurately from other viruses.

Conversely, despite the moderate accuracy of the biased-region model, this model does not seem to produce intelligent host classifications. The high true negative and false negative rates suggest that the model guesses that a virus is non-human most of the time, rather than making meaningful predictions based on different patterns of codon usage. Perhaps the model was unable to learn sufficient patterns due to the small sample size of the training dataset, or perhaps using the full genome of the virus simply provides a more complete picture of viral codon usage bias. Either way, this work finds that using the full genome codon usage is superior to working only from biased regions of the genome.

### Limitations of This Work

The major limitation of this work is sample size. We were only able to obtain genomes of 220 human-infecting viruses, and only 154 of these viruses contained sufficient biased regions of the genome to include in the biased-region classifier. As a result, our validation set was only 44 viruses for the full genome classifier and 30 viruses for the biased-regions classifier. With this small sample size, our results are somewhat influenced by random chance. If we were to train the models on a different 90% of the data and validate on the remaining 10%, our model's accuracy, false positive rate, or false negative rate may differ. However, noting that our cross-validation errors are within a reasonable range of our validation errors gives us more confidence that our results would not differ wildly due to random chance, as ten other models from within this dataset averaged out to similar accuracies.

Another limitation of this work for many viral genomes is genome size. Viruses with short genomes may not provide enough information to be certain that a particular codon usage bias is due to systemic favor of some synonymous codons over others. In such a short genome, codon usage distributions may be more strongly influenced by random chance. Genome size is especially limiting when attempting to work only from biased regions of the genome.

**Future Directions**

The success of both classifiers indicates that codon usage bias may be a useful factor to incorporate into larger models of potential viral threats to the human species. For codon usage bias to be a useful factor in threat detection, future work would need to explore codon usage bias pre- and post-zoonotic transfer to determine the extent of the predictive power of codon usage bias. We know that codon usage could be more similar in human viruses as a result of their infecting humans, since viruses have been observed as favoring synonymous mutations bringing their codon usage closer to that of their host (Bahir et al., 2005; Chen et al., 2020; Hershberg & Petrov, 2008;  Jenkins & Holmes, 2003; Mollentze et al., 2021). However, having a codon usage distribution which would be favorable for translation within human hosts may also confer an advantage to viruses during new zoonotic transfers. Thus, future work could examine whether pre-zoonotic transfer codon usage may be a useful predictor of viral success within human hosts.

Additional future work could refine the accuracy of our classifier models. We obtained our thresholds for moderate codon usage bias from Banerjee et al. (2005), but future work could experiment with the ideal effective codon usage threshold for filtering for biased regions of the genome. Future work could also explore the accuracy of classifiers when only distinguishing between human viruses and other animal viruses, rather than distinguishing human viruses from all viruses, as animal viruses are much more likely candidates for zoonotic transfer.

Finally, the creation of the codon usage dataset used in this work allows for the potential to ask many more questions regarding codon usage bias. Such future analyses could include: Is codon usage correlated between bacteria and their hosts? Do codon usage similarities correlate with known phylogenetic relationships? Or does codon usage correlate among other known ecological relationships? Our methods could also be adapted to other datasets of genomes, such as to examine evolution of SARS-CoV-2 codon usage over time.

**Additional Contributions**

See Appendices 1 and 2 for additional contributions of this work to the field of bioinformatics. Appendix 1 details the NCBI RefSeq database web scraper and its potential to be adapted into a package for other bioinformaticians to efficiently interface with the RefSeq database, accessing thousands of genomes without the need to download and analyze each manually. Appendix 2 details next directions for the codon usage dataset and its potential as a useful resource for bioinformaticians and machine learners alike. Easily accessible codon usage data presents the opportunity to explore many more hypotheses on how codon usage bias may be useful for distinguishing organisms.

Additionally, see Appendix 3 for a failed line of analysis, and an explanation of why the analysis was not fruitful. This analysis is included for the reference of the instructors grading this work, and is intended as evidence that much more work occurred behind the scenes than may be summarized here in this one paper.

## Acknowledgements

Many, many thanks go to Professors Rob Dorit (BIO 334) and Jessica Grant (BIO 335) for their guidance on this project. Their expertise was instrumental in helping me bridge the gap from data science into bioinformatics. I would also like to thank Professor Katherine Kinnaird (CSC 294) for sharing her machine learning expertise—this project would not be possible without the tremendous foundation I gained in Computational Machine Learning last spring. Thanks also goes to classmates Rose Evard, Shevaughn Holness, and Selin Kubali for their support throughout the development of this project.

## References

[1] Bahir, I., Fromer, M., Prat, Y. & Linial, M. Viral adaptation to host: a proteome-based analysis of codon usage and amino acid preferences. *Mol. Syst. Biol.* 5, 311 (2009). https://doi-org.libproxy.smith.edu/10.1038/msb.2009.71.

[2] Behura, S.K. and Severson, D.W. Codon usage bias: causative factors, quantification methods and genome-wide patterns: with emphasis on insect genomes. *Biological Reviews*. 88, 49-61 (2013). https://doi.org/10.1111/j.1469-185X.2012.00242.x.

[3] Brister JR, Ako-Adjei D, Bao Y, & Blinkova O. NCBI viral genomes resource. *Nucleic Acids Res.* 43(Database issue):D571-7 (2015). https://doi.org/10.1093/nar/gku1207.

[4] Chen, F., Wu, P., Deng, S. *et al.* Dissimilation of synonymous codon usage bias in virus–host coevolution due to translational selection. *Nat. Ecol. Evol.* 4, 589–600 (2020). https://doi-org.libproxy.smith.edu/10.1038/s41559-020-1124-7.

[5] Fuglsang A. Estimating the "effective number of codons": the Wright way of determining codon homozygosity leads to superior estimates. *Genetics.* 172(2), 1301-1307 (2006). https://doi.org/10.1534/genetics.105.049643.

[6] Gong, Y., Wen, G., Jiang, J. and Xie, F. Codon bias analysis may be insufficient for identifying host(s) of a novel virus. *J Med Virol*. 92: 1434-1436 (2020). https://doi-org.libproxy.smith.edu/10.1002/jmv.25977.

[7] Hershberg, R. & Petrov, D. A. Selection on codon bias. *Annu. Rev. Genet*. 42, 287–299 (2008). https://doi.org/10.1146/annurev.genet.42.110807.091442.

[8] Jenkins, G. M. & Holmes, E. C. The extent of codon usage bias in human RNA viruses and its evolutionary origin. *Virus Res.* 92, 1–7 (2003). https://doi.org/10.1016/S0168-1702(02)00309-X.

[9] Mollentze N, Babayan SA, & Streicker DG. Identifying and prioritizing potential human-infecting viruses from their genome sequences. *PLoS Biol.* 19(9), e3001390 (2021). https://doi.org/10.1371/journal.pbio.3001390.

[10] T. Banerjee, S.K. Gupta, & T.C. Ghosh. Towards a resolution on the inherent methodological weakness of the "effective number of codons used by a gene." *Biochemical and*

*Biophysical Research Communications*. 330(4), 1015-1018 (2005). https://doi.org/10.1016/j.bbrc.2005.02.150.

[11] Wright F. The 'effective number of codons' used in a gene. *Gene*. 87(1), 23-9 (1990). https://doi.org/10.1016/0378-1119(90)90491-9.

[12] Yi, K., Kim, S.Y., Bleazard, T. *et al.* Mutational spectrum of SARS-CoV-2 during the global pandemic. *Exp. Mol. Med.* 53, 1229–1237 (2021). https://doi-org.libproxy.smith.edu/10.1038/s12276-021-00658-z.

# Appendices

## Appendix 1:

### NCBI RefSeq Database Web Scraper

In order to efficiently calculate codon usage bias for large numbers of genomes, we coded an NCBI RefSeq database web scraper in Python.

Dependencies:

- Pandas
- BeautifulSoup4
- Biopython
- gzip
- os
- time
- requests
- statistics

The following is a walkthrough of the functionality of this web scraper:

Given a subdirectory of the RefSeq database, such as “viral,” “plant,” or “vertebrate_mammalian,” the web scraper collects all the species available for this taxon. This list of all species available within a taxon may optionally be saved to the user’s local machine, such that the web scraper may resume later without redownloading all the species names. If the user only desires to work with the genomes of a subset of the species available, the user may instead provide a newline-separated list of species in the format *Species_binomial/*. Additionally, the user may specify species to exclude from the analysis. Currently, this feature is configured to exclude species whose genomes are already collected from a previous run of the script.

Next, from these lists of species within a taxon, the web scraper iterates over each species directory, using the BeautifulSoup4 web scraping library to locate the reference genome assembly as the first directory in the “reference” (for most taxa) or “latest_assembly_versions” (for viruses) folder. The function for scraping the assembly information from the webpage also allows the option of multiple retries, in case the page load fails the first time. Many times over the course of this project, the NCBI RefSeq database mistook our repeated queries for a Denial of Service attack and blocked our access to the database. We found that adding a random wait of five to eight seconds between queries helped prevent our traffic from being blocked, though we were unable to completely prevent blockage. To account for the cooldown until our requests are allowed again, we pause our script for increasing intervals from 20 to 60 seconds until the connection succeeds or we have reached three retries. When a genome still cannot be accessed, we write the genome url to an error log for future reference.

If our connection to the NCBI RefSeq database is successful for a species, we download the compressed FASTA of the CDS regions of the genome assembly. As these genomes are downloaded and decompressed, we provide the option to save every genome to the user's local machine in a FASTA format. However, we discourage this option for large amounts of genomes due to the memory storage required. Instead, we recommend utilizing the memory-saving option to access each genome, calculate its codon usage, then abandon the genome in memory.

At this stage of the analysis, the user could substitute any desired calculation on the downloaded genome. For the purposes of our data collection, we calculate codon usage bias from the genome. We use Python's Biopython library to splice together all the individual genes into one CDS sequence of the genome, and we also produce a second DNA sequence by splicing together only the genes with Wright's effective codon usage statistic below 50. Then, we use Biopython to transcribe both the full and biased-only sequences into RNA, and proceed to calculate codon usage for these sequences.

To calculate codon usage, we first count instances of each codon in the "universal" codon chart by iterating over every triplet of nucleotides in an RNA sequence. Then, we divide uses of a codon by total uses of any synonymous codon to obtain a codon's frequency of use within its amino acid. For example, the frequency of ATT, which codes for Isoleucine, would be the total occurrences of ATT divided by the total occurrences of ATT, ATC, and ATA—the three synonymous codons corresponding with Isoleucine.

Next, we store each amino acid's frequency in a dataframe alongside metadata for the sequence such as length, Wright's effective codon usage statistic, accession information, species name, and RefSeq taxon. For sequences containing only biased regions of the genome, we include information on both the biased sequence's and full sequence's length and effective codon usage, as well as the proportion of the genome contained in biased regions (biased sequence length divided by full sequence length). This additional information allows us to disregard codon usage observations when the biased region of a genome is too small for accurate calculations.

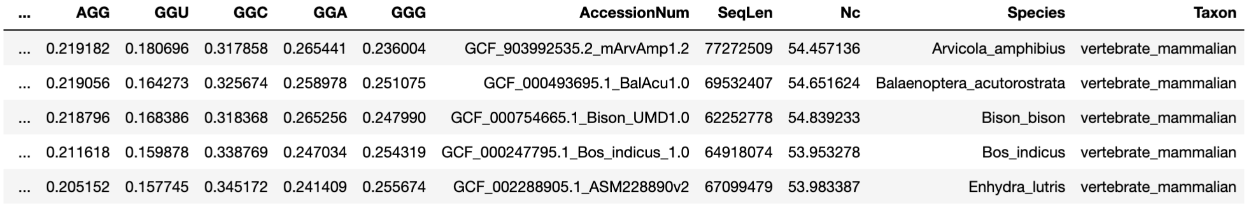
Finally, we take each dataframe corresponding with one observed codon usage distribution and store it to a working log of calculations. If the script runs to completion, it outputs a csv containing all observations in one joined dataset. However, to allow the user to stop and resume data generation without losing any data, the script also writes to a working csv file after calculating each observation. In this case, the data header is written each time to the working csv, so the supplemental *fasta_parser.py* script contains a *clean_working_csv()* function to clean the dataset to have only one header. Then, the *divide_data.ipynb* Jupyter notebook is available to publish the full and biased-only data as two separate datasets in the datasets folder, named *cub_full_genome.csv* and *cub_biased_genome.csv*.

To learn more about the web scraper code, see this project's GitHub repository (https://github.com/krhablutzel/codon-usage-bias). Now that the infrastructure exists to automatically interface with the NCBI RefSeq database and analyze each genome, this script has the potential to be adapted into a NCBI RefSeq genome web scraper package. Theoretically, the user should be able to run any analysis function on genomes as they are accessed. Such a package for interfacing with the NCBI RefSeq database would significantly increase the accessibility of wide scale calculations across numerous reference genomes.

## Appendix 2:
### Publishing an Accessible Database of Codon Usage Bias

While this work focused on a narrow segment of the NCBI RefSeq database (namely, human viruses and a random sample of other viruses), our work collected codon usage bias for thousands of other plants and animals as well. The data collection process takes numerous hours, so collection is still ongoing to collect more animal, plant, and viral genomes, and to expand the dataset to other taxa as well. As this dataset is generated, it will be published to GitHub in the project repository's *datasets* folder.

As far as we know, no accessible database of codon usage bias currently exists. Access to this dataset will allow future bioinformaticians to more easily explore additional applications of codon usage bias. Additionally, this dataset will be useful to machine learners for testing clustering and classification algorithms.



| ... | AGG | GGU | GGC | GGA | GGG | AccessionNum | SeqLen | Nc | Species | Taxon |
|---|---|---|---|---|---|---|---|---|---|---|
| ... | 0.219182 | 0.180696 | 0.317858 | 0.265441 | 0.236004 | GCF_903992535.2_mArvAmp1.2 | 77272509 | 54.457136 | Arvicola_amphibius | vertebrate_mammalian |
| ... | 0.219056 | 0.164273 | 0.325674 | 0.258978 | 0.251075 | GCF_000493695.1_BalAcu1.0 | 69532407 | 54.651624 | Balaenoptera_acutorostrata | vertebrate_mammalian |
| ... | 0.218796 | 0.168386 | 0.318368 | 0.265256 | 0.247990 | GCF_000754665.1_Bison_UMD1.0 | 62252778 | 54.839233 | Bison_bison | vertebrate_mammalian |
| ... | 0.211618 | 0.159878 | 0.338769 | 0.247034 | 0.254319 | GCF_000247795.1_Bos_indicus_1.0 | 64918074 | 53.953278 | Bos_indicus | vertebrate_mammalian |
| ... | 0.205152 | 0.157745 | 0.345172 | 0.241409 | 0.255674 | GCF_002288905.1_ASM228890v2 | 67099479 | 53.983387 | Enhydra_lutris | vertebrate_mammalian |

Figure A1: A sample of the data available in the codon usage bias dataset.

# Appendix 3:

## Clustering and Classifying Mammals, Other Vertebrates, and Invertebrates

This line of analysis was discarded after discovering multiple flaws. First, we discovered that our two observed clusters are based purely on whether a genome uses tryptophan, and realized that we should exclude tryptophan from our analysis (since tryptophan, like methionine, has only one corresponding codon). Then, we realized that some genomes excluded tryptophan because our dataset had only calculated codon usage for the first 100 codons in a genome, due to a bug in our codon counting function. After fixing our codon counting function, we have recollected our codon usage dataset, and this analysis is no longer accurate nor relevant. However, for purposes of showcasing both failed and successful lines of analysis in our final report, we include a walkthrough of this analysis process below.

During our initial data exploration, we visualized codon usage variation in two dimensions using principal component analysis. We observed slight differences in codon usage among mammals, other vertebrates, and invertebrates; however, a more curious aspect of the data was that it seemed to fall into two distinct clusters (Figure A2).

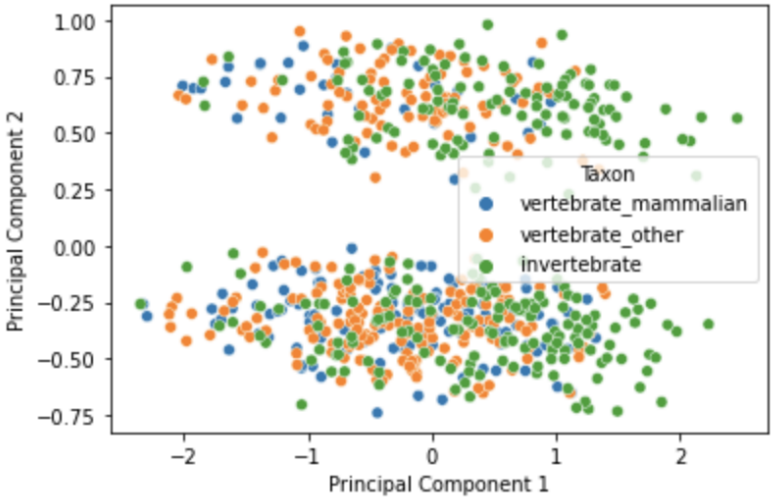


Figure A2: Animals fall into two very distinct clusters of codon usage. These clusters do not appear to correlate with known taxonomic boundaries.

We applied a k-Means clustering algorithm to our data to see if the algorithm could learn the difference between these two clusters, even if there is no immediate explanation for the clusters. Even more curiously, the clustering algorithm failed to find the two clusters (Figure A3).

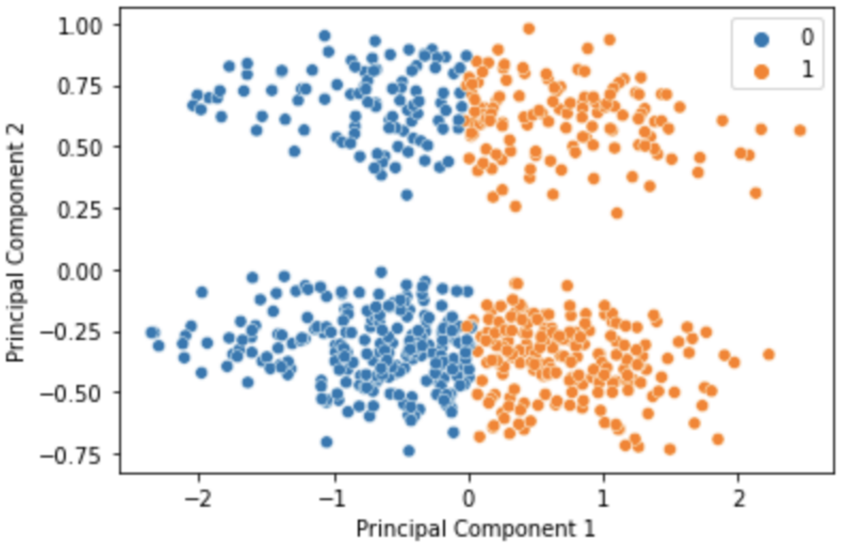


Figure A3: The labels applied by the k-Means clustering algorithm, 0 and 1, fail to describe the clusters which are very visually apparent in our data.

After plotting the distributions of each codon's frequencies, we discovered that UGG (Tryptophan) acts as a categorical variable, only taking values of 0 or 1 (Figure A4).

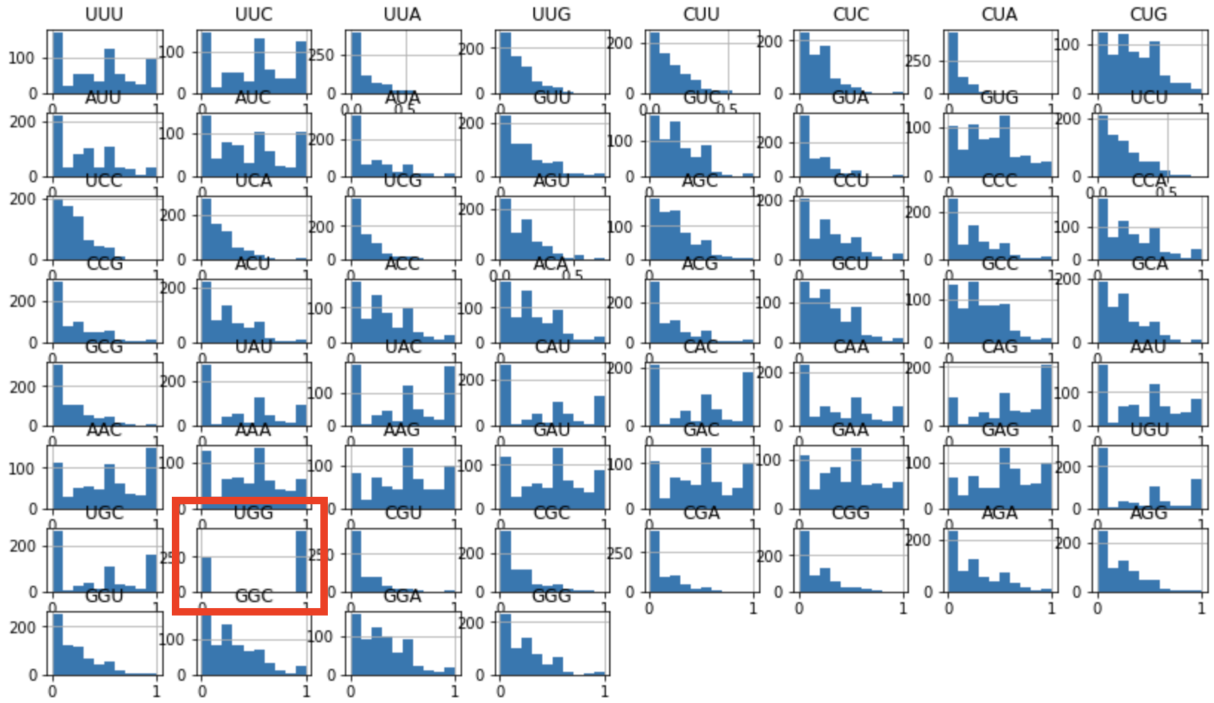


Figure A4: The codon usage distribution of UGG (Tryptophan) is highlighted in red.

If we repeat our dimension reduction excluding the tryptophan codon in addition to the methionine and stop codons, we no longer observe two clusters of data (Figure A5).

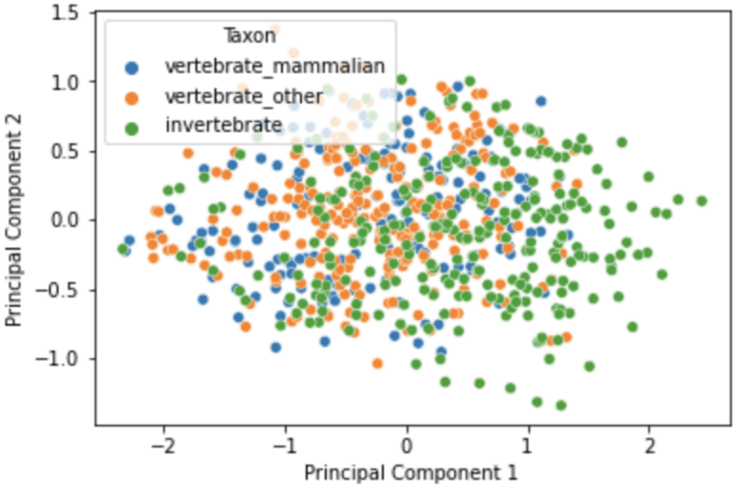


Figure A5: When we exclude tryptophan, codon usage among animals forms a singular cluster.

This line of exploration serves as a reminder of why we exclude tryptophan in addition to methionine and the stop codons when analyzing codon usage. With only one codon, tryptophan does not provide useful information for our analysis.

Finally, it seemed a bit curious that tryptophan was never used in some genomes. In fact, looking at the distributions of each codon's usage frequency (Figure A4), many codons appeared to never be used. We solved this puzzle when we discovered that our codon counter was only iterating over the first 100 codons in the genome, rather than the whole genome—a relic of testing during code development. Once we fixed this bug, tryptophan appears in over 99% of genomes collected.