**Open Reading Frame Prediction of the Porcine Reproductive and Respiratory Syndrome Virus by Algorithmic Approaches**

**Amos Cameron, Heffel G. Matthew**

**Abstract**

In many RNA viruses the viral genome is composed of a single strand of nucleic acid containing multiple open reading frames. Open reading frames, ORFs, are coding segments of the genome that are translated from RNA into protein. They are marked by a start codon, “AUG”, and end once a stop codon is reached in the same reading frame. Specifically, Porcine Reproductive and Respiratory Syndrome Virus’s genome is a single stranded RNA with multiple open reading frames. There are however significantly more start codons in the genome than just the ones that initiate translation. The problem being addressed here is why some of these start codons are being selected to initiate translation and the others are being passed over as if they were any other codon.

**Introduction**

Porcine Reproductive and Respiratory Syndrome Virus is a member of the Group IV viruses (positive sense single stranded RNA), the order *Nidovirales*, the family *Arteriviridae*, and the genus *Arterivirus*.It is the causative agent for Porcine Reproductive and Respiratory Syndrome. The symptoms of this virus are reproductive failure in sows as well as respiratory failure in newborn piglets. The virus has two main strains, the North American strain and the European Strain which differ at the genomic level by roughly 40 percent. This virus has a high rate of mutation so much to the point that pigs that have been vaccinated for this virus are still falling ill to the disease because new versions of the virus keep evolving. Globally, this virus is costing the agriculture industry roughly 2.4 billion dollars each year, 600 million from the United States and 1.8 from Europe.

The PRRSV genome contains 9 open reading frames. ORF1a/b, ORF2a/b, ORF3, ORF4, ORF5, ORF6, and ORF7. All of the open reading frames begin with ribosomal binding to the RNA at a start codon except for ORF1b which is the result of is a -1/+2 ribosomal frameshift during translation causing the normal start codon to be missed and the ORF1a protein to be extended until a stop codon is met in the new reading frame. ORF1a/b is a nonstructural protein and is involved in genome replication, ORFs 2-5 are glycosylated membrane proteins, ORF 6 is a matrix protein, and ORF 7 codes for the nucleocapsid.

By running a well-developed ORF prediction tool on an unannotated genome not only would could genomes be automatically annotated for a researcher, but also new open reading frames that resulted from mutations in the virus may potentially be discovered.

**Methods**

The approach to solving this problem is mainly by using algorithmic methods to recognize patterns within the genome. The first step in this process was to gather the data set. The dataset was gathered by pulling every file from GenBank that matched the tag “PRRSV” and was submitted before 2017. The a previously developed python script was used to ensure that the data in the file is in the correct format as well as of the correct nature. Using this script, data was removed that was either nonuniform in format or not of the correct species. The removed data was placed and can be found in the project directory “diffData,” as to not remove data without record. Once all bad data was removed it was time to move on to developing the algorithm.

The first step in developing the prediction algorithm was to parse through each file and index, based on the publisher’s annotation’s, the start codons that do initiate translation. From these indexed positions, sequences were generated by taking 10 nucleotides upstream as well as 12 nucleotides downstream of the start codon, resulting in sequences of length 25 that were then stored as strings and placed in an array of sequences representing true start codons. For testing purposes and to develop scoring thresholds, the entire genome was also iterated through, doing the same process for every start codon found, adding the sequences that were not from a true start codon to a list representing false start sequences.

The next step was to develop a scoring matrix. This was done by fist initializing an empty matrix with the dimensions 4x25, 4 rows representing each of the four nucleotide possibilities as well as 25 columns representing each position in the true start codon sequences. Then the list of true start codon sequences is iterated through and for each sequence the individual nucleotides in each position are checked and the value representing the identified nucleotide at its corresponding position in the matrix is increased by 1. The result is a 2D matrix with each column containing the number each type of nucleotide in that position. This matrix is then converted into a frequency matrix by dividing each entry in the matrix by the total number sequences in the true start codon list. This new matrix contains the frequency of each nucleotide at each position rather than the total number.

This matrix is then used as an input to a scoring function along with the sequence to be scored the sequence is scored by setting an initial score variable to 1 and then comparing the nucleotide at each position of the sequence with that position’s column in the matrix and multiplying the score variable by the frequency of the given nucleotide at that position, effectively giving the positions surrounding the start codon different weights of importance based on frequency of occurrence.

Originally the idea was that this scoring method would be able to accurately predict whether any given start codon and its surrounding nucleotides are responsible for initiating translation or not. After testing eliminating overlap between the low scoring true start sequences and the high scoring false start sequences was not possible, meaning open reading frames could not be accurately predicted in this manner, so an innovative approach was taken.

Instead of using all known open reading frames to build the scoring matrix, only ORF1a from each annotated genome was used as input to build the scoring matrix. True ORF1a start codons as well as false start codons throughout the input genome dataset were located and scored. The results had no overlap between the lowest scoring true start codon and the highest scoring false start codon meaning that the start site for ORF1a/b can be annotated with 100 percent accuracy. The score threshold for ORF1a was evaluated to be the lowest true score from ORF1a plus the highest false score from the input genomes divided by 2.

The approach for identifying the remaining open reading frames in the genome is slightly more complex, using another scoring matrix as well as knowledge of the viral genome. The new scoring matrix was built using all of the annotated open reading frames from the input genomes except for the ones annotated as ORF1a. Using this scoring matrix as the input for the scoring functions, all true start codons other than 1a and all false start codons from the input dataset were located and scored. The results showed that true and false start codons against this scoring matrix still generate an overlap of the lowest true start score and the highest false start score, but less of an overlap than was seen in the previous method. This scoring method is next employed in a dynamic construction of a genome’s open reading frames using what is known about the viral genome.

ORF1a is found by using the above-mentioned method of a scoring matrix built off of only ORF1a input and then from the beginning of the genome scoring each start codon until one is found with a score above the set scoring matrix’s ORF1a score threshold. The found start codon is the beginning of ORF1a. The end of ORF1a is then found by looking for the first stop codon in the reading frame, the first stop codon found is the end of ORF1a. ORF1a is annotated. Since ORF1a and ORF1b start at the same location, to annotate ORF1b the start site of ORF1a is taken as the start site of ORF1b and then the end of ORF1b is found by looking for the first stop codon after the end of ORF1a in a -1 reading frame because of the ribosomal frameshift.

ORF2a is found by starting at the end of ORF1b and checking every start codon until one is found whose score is above the threshold score for its scoring matrix. This is the start site of ORF2a. The end of ORF2a is marked as the first stop codon in the reading frame starting at the beginning of ORF2a. ORF2b is found by the same method starting immediately after the start codon for ORF2a.

ORF3 is found by only a slightly different method. Since it is known that ORF3 has a minor overlap with the end of ORF2a, ORF3 is found by taking the end of ORFa2 and subtracting a percent of the length of ORF2a. Moving forward form that position, scoring each start codon that is passed pass, the first start codon with a score above the threshold is identified as ORF3. The end of ORF3 is marked as the first stop codon from the start of ORF3 in the same reading frame. ORFs 4 and 5 are identified by an even more dynamic method.

ORF4 is known to have a slight overlap with the end of ORF3, but if searching begins too far back into ORF3 the wrong start codon will be identified. So, the end index of ORF3 is taken instead and a percent of ORF3’s length is subtracted. Then moving downstream, scoring each start codon until one with a score above the threshold is found or a certain distance was covered. If the distance capacity is reached before the start site of ORF4 is found, the backtrack value from the end of ORF3 is incremented and the process is repeated. This continues until ORF4 is identified. The end of ORF4 is found by the same mechanism as the ends of the rest of the ORFs other than ORF1b. The remainder of the ORF endpoints will be found by the same method. ORF5 is found by a similar mechanism. ORF5 only sometimes overlaps with the end of ORF4 so codons just past the end of ORF4 are scored in order and if one is found with a score above the score threshold before a small length, this is labeled as ORF5’s beginning. If nothing is found in the small segment, a small traceback into ORF4 is taken and the process is repeated forwards.

ORF6 barely overlaps with the end of ORF5 so codons beginning being scored just before the end of ORF5 and the first one found with a score above the threshold is identified as the start site of ORF6. ORF7 is found in the same was ORF6 except the endpoint of ORF6 is used rather than ORF5.

**Results**

Tested against the input dataset as well as genomes not used to build the scoring matrices, inputting the raw genomes and comparing the prediction algorithm’s output to the genome’s published annotations, the prediction algorithm correctly identifies ORFs 1 through 7 including 1b and 2b for 98 percent of the genomes. The remaining 2 percent were however, not complete failures. 100 percent of the genomes run through the algorithm were correctly annotated for ORF1, but at some point between ORF2 and ORF7 an ORF is wrongly identified and based on the sequential nature of how the algorithm predicts open reading frames, every subsequence ORF after the first incorrectly labeled ORF is incorrectly labeled as well, with the exception of a mislabeling on ORF2b. This mechanism causes an exponential increase in the probability of incorrectly predicting an ORF as the number of ORFs being predicted increases. The 2 percent of data that had an occurrence of misprediction generally had its first wrong annotation at ORF5 or 6.

A 25 base pair consensus sequence for ORF1 initiation was developed as CCCUUUAACCAUGUGGGAUACUU, which when blasted, matched with 8,945 species of PRRSV. The consensus sequence developed from the remaining ORFs is CUGAUCAGAAAUGGCGUCGAACUUU which when blasted only matches with 2101 species of PRRSV. This gives us some explanation as to why ORF1 can be predicted with perfect accuracy and without knowledge for the genome, but the remaining ORFs cannot.

**Discussion**

A byproduct of the tool that has been built is the GenBank flat file parser, that was the first line of defense in omitting bad data, could be used by NCBI on newly submitted genomes to check for proper formatting by the publisher. This feature would not be restricted to only genomes of Porcine Reproductive and Respiratory Syndrome Virus but could work on any GenBank flat file as it is only a matter of formatting being checked and has nothing to do with the specific species.

The developed tool allows a researcher to annotate a raw PRRSV genome’s open reading frames with high accuracy. In addition to annotating raw genomes, when results were tested against partially annotated genomes the prediction algorithm was noticed to be filling in the open reading frames that were not mentioned on GenBank. The original interpretation was that this was an error because the values being returned back were not in published annotations. It was later discovered that this was not a mistake on the algorithm’s part, but rather a lack of original annotation by the publisher. So, this tool could be used to offer suggested open reading frame annotations on older genomes submitted to GenBank.

Regarding the 2 percent error rate, a few theories have been developed as to why these error cases arise. One being that some of the genomes being tested are more distanced phylogenetically than others, causing the “genome knowledge” attributes of the algorithm to not quite be correct. It is known that some strains of the viral species can be more than 40 percent genetically different. Another possibility for the error cases has to do with the biochemistry of the ribosome binding to the RNA in translation initiation. The virus does not translate itself with its own machinery, just like many other viruses, instead relying on the host ribosomes to do the work. This means that if two pigs, the hosts, have slightly different ribosomes, it may cause a difference in the way the RNA is translated, resulting in deviations from expected results.

With further development of the algorithm, including machine learning approaches as well as knowledge of other genomes and larger sets of training data, this algorithm could be improved to perform the same function on other viral species or perhaps even any multi-ORF RNA. In addition to being adapted to perform the predictions on other species, the prediction mechanics for PRRSV open reading frames could be fine tuned for more accurate predictions.

**Code**

All code for this tool was developed in python. The pandas data frame python module was used as well. The project directory is represented as follows. The archive directory stores old code generated in the process of completing this project but is no longer needed for the function of the tool. The “diffData” directory contains all of the GenBank flat files removed from input data as bad data. The “inputGenomes” directory contains all of the GenBank flat files used as input to develop the scoring matrices. The “loadData” directory stored all necessary files generated by the script “buildMatricies.py” and used by “Annotate.py” to annotate genomes. The “testGenomes” directory contains fasta files that the “Annotate.py” script with predict and annotate open reading frames for. This is where a genome of interest’s fasta file would be put in order to annotate it. The python script “buildMatricies.py” uses the input genomes dataset to develop scoring matrices and score thresholds for use in annotation. This script takes a while to run, but only needs to be run once. The python script “Annotate.py” used the data generated by “buildMatrcies.py” to annotate genomes in the “testGenomes” directory.

**References**

Benson, Dennis A. et al. “GenBank.” *Nucleic Acids Research* 33.Database Issue (2005): D34–D38. *PMC*. Web. 4 May 2018.

Wes McKinney. **Data Structures for Statistical Computing in Python**, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

**Author’s Contributions**

Cameron aided in the code by programming some helper scripts, but mostly the code and algorithmic approaches were done by me. The project scope and goals were defined by me. Data and output testing was done by both Cameron and I. The presentation slides were made by Cameron and the reports were done mostly individually with structure and outlines shared.