* Introduction
* Motivation
  + Spliceosome is an important RNP for genetic expression. [6], [7]
    - Large RNP structure composed of snRNP that bind to introns to remove them.
    - Removes Introns and combines Exons into pre-mRNA.
      * Segmentation of genes allows for small number of genes to provide wide variety of proteins.
      * There are two types of Introns – U2 and U12 – based on the snRNP that binds to them. [6]
      * ~700 U12 introns.
    - U1 binds to RNA, U2 binds to RNA, recruits U4, U5, U6, bends RNA and splices out the introns. [6]
      * Numerous proteins that help facilitate this.
      * U1 binds to 5’ ss.
      * U2 binds to branch point.
        + Branch point is located ~40 nucleotides upstream from 3’ ss.
      * U4, U5, and U6 are recruited to the intron.
      * snRNPs bend the intron around, splice it out from the exon, and combine the exons to form mRNA.
    - Alternative splicing makes this even more challenging.
      * A spliceosome can *exclude* some exons in different patterns.
  + Splicing errors, while rare, can result in cancer and other disease states. [11]
* Background
  + On average a gene has 8 exons and 7 introns [1], [2].
    - Average Size of:
      * Exons: 160 bp [see if std is in one of those sources]
      * Introns: 5800 bp [again, see if std is there]
  + GT-AG is the canonical splice site.
    - Looking at 2-mer occurrences, GT is below average.
    - AG is more common, meaning there are several false 3’ splice sites.
    - Relying simply on GT-AG would yield many false introns.
  + High fidelity of the Spliceosome remains a mystery.
    - Many factors play into exon-intron identification.
      * Include but not limited to:
        + [Canonical] Splice Site
        + RNA secondary structure [9]
        + Branch site identification.
        + Exon-Complex Definition
* Methods
  + Assumed the spliceosome worked locally and only on some sequences [3].
  + Bulit a 1D model and a 2D model in Tensorflow.
    - Used Tensorflow as it is a robust and well supported Machine Learning Library in Python
      * Has several visualization outputs.
      * Incorporates other ML libraries as well, such as Keras, offering additional features.
  + Collected data from UCSC Genome.
    - 200 genes per chromosome.
      * 4800 genes used.
    - Took their exon and intron regions and turned them into numerical representations.
      * 20,000 exons, 25,000 introns
      * Additional introns were used due to selective requirements.
      * Due to the need for sufficient numerical representation, several shorter exons were discarded.
  + 1D treated the sequence as a time series analysis.
    - A natural and intuitive method for examining the genome.
      * The results were poor.
    - Used one hot encoding to represent the data numerically.
      * A = [1, 0, 0, 0], C = [0, 1, 0, 0], G = [0, 0, 1, 0], T = [0, 0, 0, 1]
    - Used entire region sequence up to 1000 nucleotides.
      * + Buffered shorter sequences with 0’s.
  + 2D used Chaos Game Representation to bring genetic data is higher dimensions. [4]
    - An example of using high dimensions to see structure: Lorenze Attractor
      * Data appears random/pattern-less.
      * Moving to higher dimensions reveals a pattern that can be understood.
    - Used a Convolution Neural Network to build the model.
      * CNN is a well-studied and commonly used method for image analysis in ML.
      * Allows images to be rotated and flipped while still being able to identify underlying structures that make up the image.
      * Several well understood techniques could be adopted to provide better loss functions and handle overtraining.
    - CGR: Sierpinski Triangle with 4 vertices and uses nonrandom data (the genetic sequence) to generate points.
      * Divide the image into P bins, where P = 4\*\*k.
      * Each bin represents the frequency of a k-mer motif in a given region. [8]
    - Generates a Fractal by counting the frequency of a k-mer motif.
      * Used 6, 7, 8, 9-mers
      * Most used were 6-mer.
        + Motivated for biological reasons: GT-AG is recognized +9 and -6 from the SS [3]
    - Motivated by assuming self-similarity within Exons and self-similarity within Introns.
      * Separating Exons from Introns due to their separate fractal dimensions.[14]
    - GT-AG is part of the structure, but the interior structure is also important.
    - Used two methods for creating the images.
      * First method:
        + Sequence had to be >100 nucleotides long.
        + Had more introns than exons.
      * Second Method: Histogram Method
        + This was done to check if length of introns were impacting the results.

Introns are much longer than Exons: was the additional data points skewing the results?

* + - * + Used only introns that were between the longest and shortest exon of the same gene.
        + Most exons were between 10-1000 nucleotides long.
        + Most introns were closer to 1000 nucleotides in length.
        + Had more exons than introns.
* Results
  + 1D results could identify training data very well, but could not validate.
    - Appeared to just randomly assign classification of validation.
    - Over trained the data.
  + 2D results were able to train and validate very well.
    - Best results were using the histogram method.
      * Suggests length is not the only factor of determining exon vs intron.
    - Looked at 6-mer predominantly.
    - Training accuracy ~95%, validation accuracy ~92%
      * 9-mer was slightly worse.
      * Looked at 7 and 8-mer, results will somewhat similar to 9-mer, suggesting that 6-mer was the optimal choice.
* Additional Analysis
  + Looked at “Time Embedding”
  + Considered a forward and backwards window
    - where
      * Semi arbitrary choices: AG are Purines and CT are Pyrimidines.
      * Wanted to group Pu and Py and keep the nucleotide they bond to numerically distant from each other.
  + Found a substantial “void” for
    - The large tracks of limited nucleotides lead in and out of that void.
  + **Need to do machine learning on these plots**
  + **Need to cross reference the void with known U1 binding sites**
* Future Work
  + Still need work in identifying specific points.
    - This looks at the underlying structure, but the Spliceosome brings a scalpel as well.
  + Does not address alternative splicing.
    - This does not explain how some.
* References
  + [1] Patterns of exon-intron architecture variation of genes in eukaryotic genomes, Zhu, Zhang, Zhang, Yang, Chen, Tian
  + [2] Intron-exon structure of eukaryotic model organisms, Deutsch and Long
  + [3] Pre-mRNA Secondary Structures Influence Exon Recognition, Hiller, Zhang, Backofen, Stamm
  + [4] Chaos Game Representation and its Applications in Bioinformatics, Lochel, Heider
  + [5] A novel role of U1 snRNP: splice site selection from a distance, Singh, Singh
  + [6] Spliceosome Structure and Function, Will, Luhrmann
  + [7] A Day in the Life of the Spliceosome, Matera, Wang
  + [8] further citation on CGR. I know there’s another paper on the k-mer bin representation, I just need to find it. It’s ‘91 or ‘92 I think?
  + [9] Thing about RNA secondary structure
  + [10] Thing about U1 snRNP
  + [11] Aberrant RNA Splicing in Cancer and Drug Resistance, Wang & Lee
  + [12] Spliceosome book (or article out of it)
  + [13] Intron exon boundary junctions in human genome have in-built unique structural and energetic signals
  + [14] The fractal Nature of RNA Secondary Structures

