Mengyi Liu
Yiyao Jin
Shahrzad Nikkhah
BENG 182
Project 4 Report

Frameshift Mutation Rescue

Background:

Frameshift mutation occurs when the addition or loss of DNA bases changes a gene's reading frame. The shift of the reading frame can change the codons for amino acids. This can result in a change in the genetic message, thus resulting in completely different translation. Frameshift mutation not only changes the reading codons to code for a different amino acids, but also alters the first stop codon encountered in a sequence.

Transcription and translation are responsible for making proteins by reading the information from the DNA. In general, frameshift mutation causes error in the amino acids, the primary structure of a protein; this bug carries on when the protein folds and gets to its quaternary or functional structures. Often times, this error makes the protein lose its function and eventually causes disease.

To rescue the frameshift mutation, there is a process called alternative splicing that happens in human genome. The majority of multi-exonic genes in human are alternatively spliced and the most common mode of the alternative splicing is exon skipping, which helps the cellular machinery to skip over an exon. We will utilize this machinery in our project.

Method:

The goal of this project is to identify the locations in the exons of genes in the human genome, where frameshift mutations can be rescued by alternative splicing. The method we used to rescue the frameshift mutation in this project involves three steps.

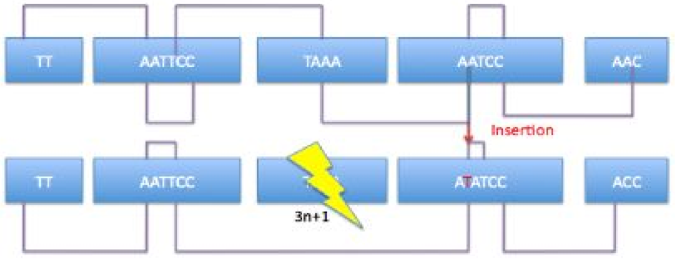
- In the first step, we get all the indices of translated exons from our dataset and remove the UTR region.
- For the second step, we get all the sequences of the translated exons by using Pysam, a Python library to get sequence from fasta file. We also found the reverse complement of those sequences on the negative strand.
- In the third phase, we suppose there is an insertion or deletion of one or two nucleotide at a position in an exon. Then we did case analysis for rescuing the mutation.

To rescue the mutation by using alternative splicing (exons skipping), we considered four different cases: insert one or two nucleotides, or delete one or two nucleotides. The difference between these cases lies in the length of the spliced exon (previous or next exon of the mutated exon).

For an example, in the graph below “Example illustration 1”: We suppose that one nucleotide got inserted to our original DNA. Then we check if the length of the previous or next exons is 3n+1. If any of the exons has such a length, then skipping this exon might rescue the mutation. However, we still need to check if any new stop codon would be created by splicing this specific exon. Only when there’s no stop codon created, can we conclude that skipping this exon can rescue the frameshift mutation.

There are lots of subcases we had to consider (72 = 2*4*9 cases in total) for checking the possible sequences that encode new codons after a frameshift mutation. Starting with 2 cases for rescuing either by deleting the previous exon or the next exon, each case has 4 subcases: inserting 1 or 2, or deleting 1 or 2 nucleotide(s). Each subcase further splits into 9 cases by considering the all the combinations of the exon reading frames and relationship between mutation positions and the reading frame of the mutated exon.

Example illustration 1: Insert 1 nucleotide



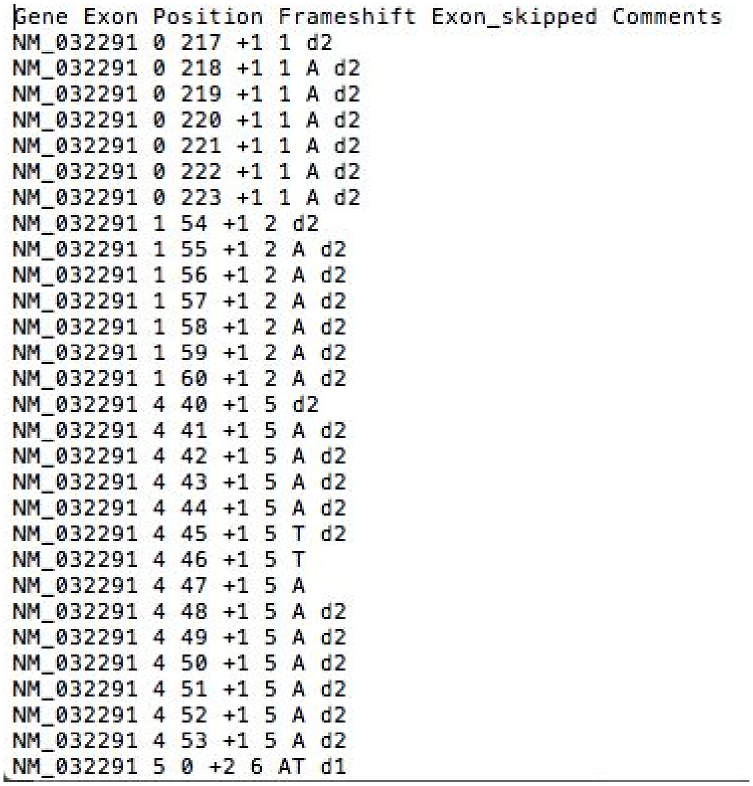
The original frame: TTA, ATT, CCT, AAA, AAT, CCA
The Frame after splicing : TTA, ATT, CCA, TAT, CCA

Tools used:

- Jupyter Notebook
- Python, and Python libraries including Pysam, Pandas, etc.

Output table:

Here's the beginning of the output table (please see the Project Code for the entire file). In this table, any singleton nucleotide or two nucleotides indicates an insertion of that specific sequence. “d1” or “d2” in the comments section means deletion of one or two nucleotides (respectively) that causes the frameshift mutation. This output table includes all the possible rescue methods, indicated by skipping which exon, given a specific frameshift case for a certain position on an exon of a certain gene.



```
Gene Exon Position Frameshift Exon_skipped Comments
NM_032291 0 217 +1 1 d2
NM_032291 0 218 +1 1 A d2
NM_032291 0 219 +1 1 A d2
NM_032291 0 220 +1 1 A d2
NM_032291 0 221 +1 1 A d2
NM_032291 0 222 +1 1 A d2
NM_032291 0 223 +1 1 A d2
NM_032291 1 54 +1 2 d2
NM_032291 1 55 +1 2 A d2
NM_032291 1 56 +1 2 A d2
NM_032291 1 57 +1 2 A d2
NM_032291 1 58 +1 2 A d2
NM_032291 1 59 +1 2 A d2
NM_032291 1 60 +1 2 A d2
NM_032291 4 40 +1 5 d2
NM_032291 4 41 +1 5 A d2
NM_032291 4 42 +1 5 A d2
NM_032291 4 43 +1 5 A d2
NM_032291 4 44 +1 5 A d2
NM_032291 4 45 +1 5 T d2
NM_032291 4 46 +1 5 T
NM_032291 4 47 +1 5 A
NM_032291 4 48 +1 5 A d2
NM_032291 4 49 +1 5 A d2
NM_032291 4 50 +1 5 A d2
NM_032291 4 51 +1 5 A d2
NM_032291 4 52 +1 5 A d2
NM_032291 4 53 +1 5 A d2
NM_032291 5 0 +2 6 AT d1
```

Summary and Discussion:

There are around 66% of exons that can be rescued. This is a decent percentage of exons, assuming that we could splice any previous or next exon of the mutated exon.

Annotation of frameshift mutations:

We compared the output we got and the actual frameshift mutations that happen in genes of human. An example of a such an overlapping frameshift mutation is on the 5th exon of gene ADC, at position 33549670. We found that if there is a single nucleotide deletion, we could skip the next exon (6th exon) on this gene to rescue. The variant database shows that there is a frameshift mutation that happens at this position, where GC changes to a G. This verifies our output “single nucleotide deletion”.

References:

1. “What kinds of gene mutations are possible?”

   https://ghr.nlm.nih.gov/primer/mutationsanddisorders/possiblemutations

2. Variant database, http://exac.broadinstitute.org