Frameshift Mutation Rescue

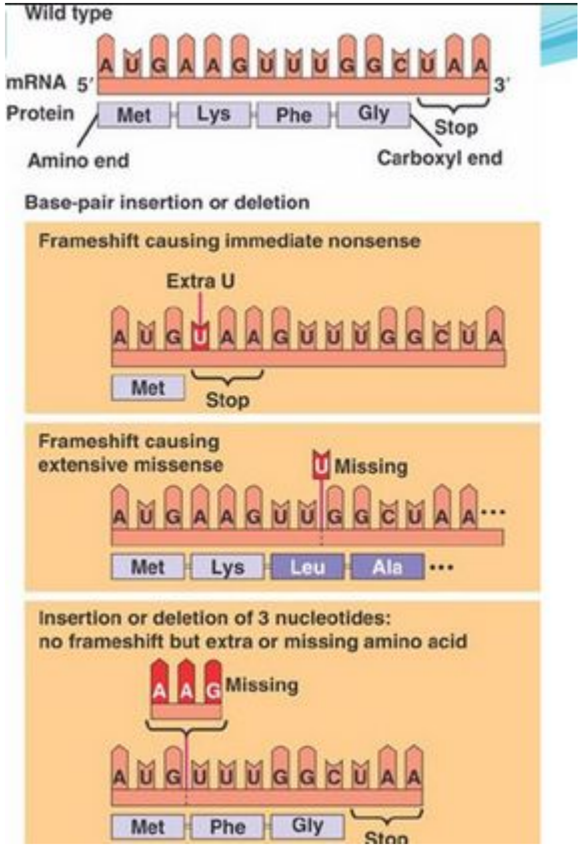
Miko (Mengyi Liu), Yiyao Jin, Shahrzad Nikkhah

Background

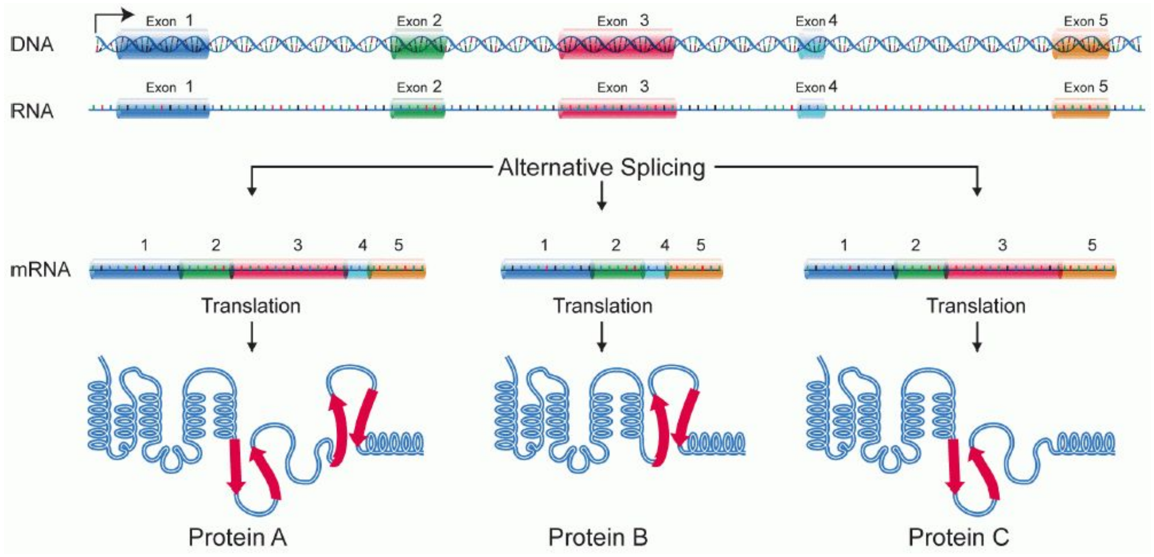
Frameshift mutation

- Caused by the insertion or deletion of nucleotide(s)
- Can cause nonsense, extensive missense, or the insertion/deletion of single amino acids
- Why would a frameshift mutation cause a protein to lose its function?

- If primary sequence is wrong , then sequence will also be wrong ---> Shape changes ---->function lost


Wild type
mRNA 5′
A U G A A G U U U G G C U A A
3′
Protein
Met
Lys
Phe
Gly
Stop
Amino end
Carboxyl end
Base-pair insertion or deletion
Frameshift causing immediate nonsense
Extra U
A U G U A A G U U U G G C U A
Met
Stop
Frameshift causing extensive missense
Missing
A U G A A G U U G G C U A A
Met
Lys
Leu
Ala
Insertion or deletion of 3 nucleotides: no frameshift but extra or missing amino acid
A A G
Missing
A U G U U U G G C U A A
Met
Phe
Gly
Stop

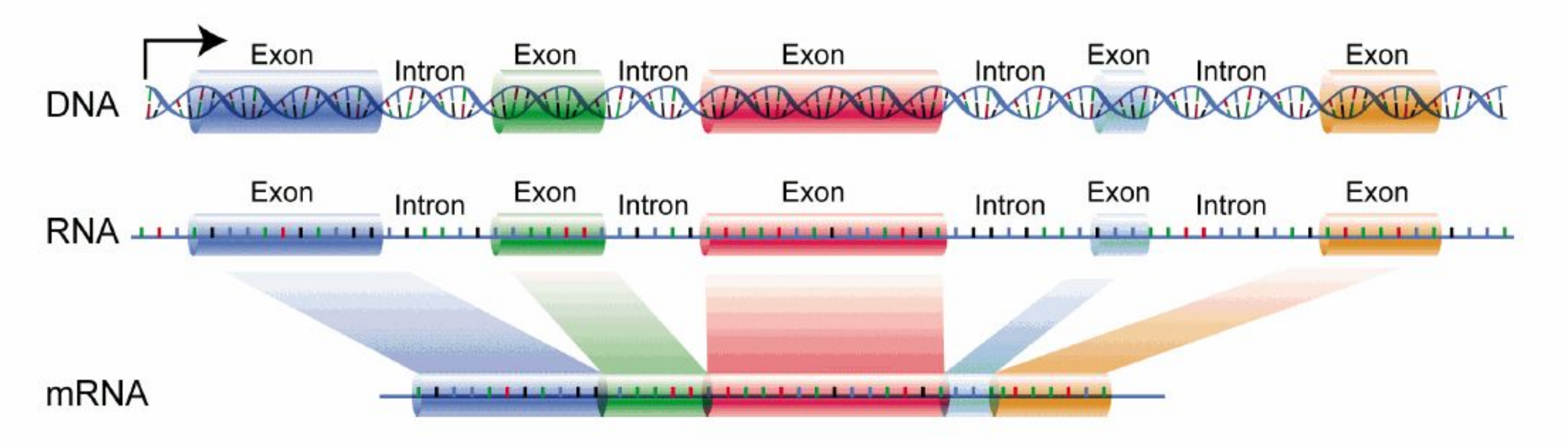
Alternative Splicing



- In human, ~95% of multi-exonic genes are alternatively spliced.
- Most common mode of alternative splicing: **Exon skipping**

Goal

- Identify locations in the **exons** of genes in the human genome, where frameshift mutations can be **rescued** by alternative splicing.



Input Data

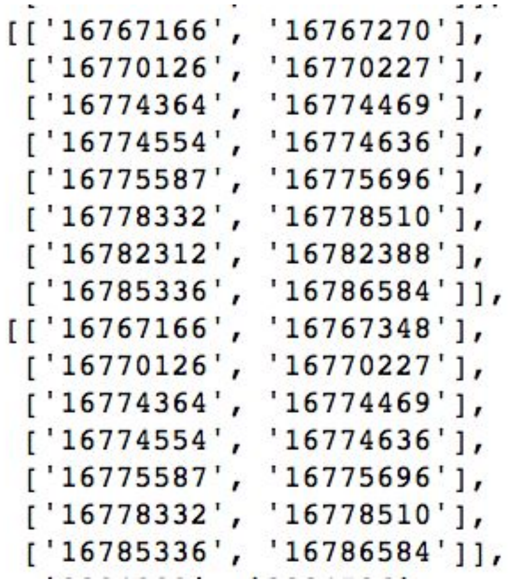
- RefSeq **transcript file**
  - Name of the gene
  - Chromosome name
  - Strand (+/-)
  - Transcription start and end points
  - Translation start and end points
  - Number of exons
  - Exon start and end points
  - Reading frames of each exon
  - ...

- Human reference genome sequence
  - **FASTA** file
  - A text-based format for representing either nucleotide or peptide sequence

Methods

STEP 1: Get all the <u>indices</u> of translated exons

- remove the UTR region

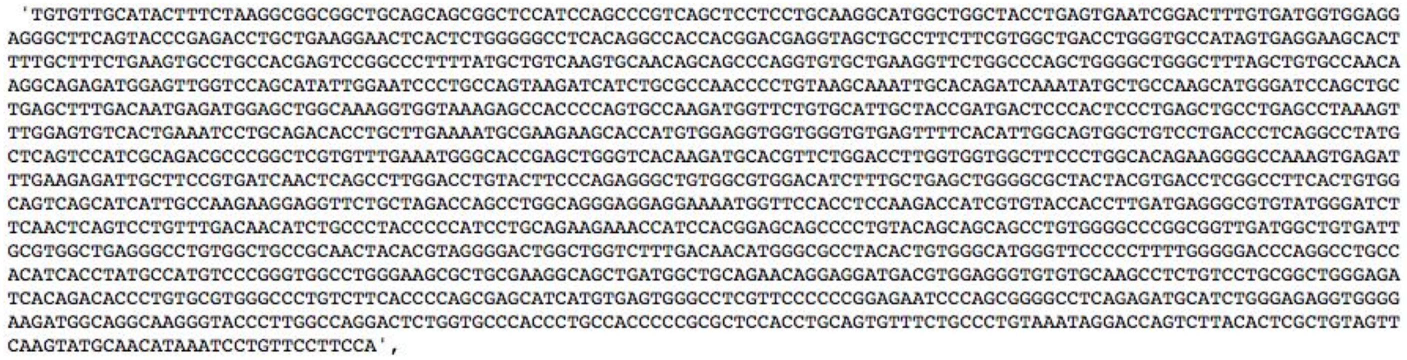


```
[['16767166', '16767270'],
 ['16770126', '16770227'],
 ['16774364', '16774469'],
 ['16774554', '16774636'],
 ['16775587', '16775696'],
 ['16778332', '16778510'],
 ['16782312', '16782388'],
 ['16785336', '16786584']],
[['16767166', '16767348'],
 ['16770126', '16770227'],
 ['16774364', '16774469'],
 ['16774554', '16774636'],
 ['16775587', '16775696'],
 ['16778332', '16778510'],
 ['16785336', '16786584']],
```

Methods

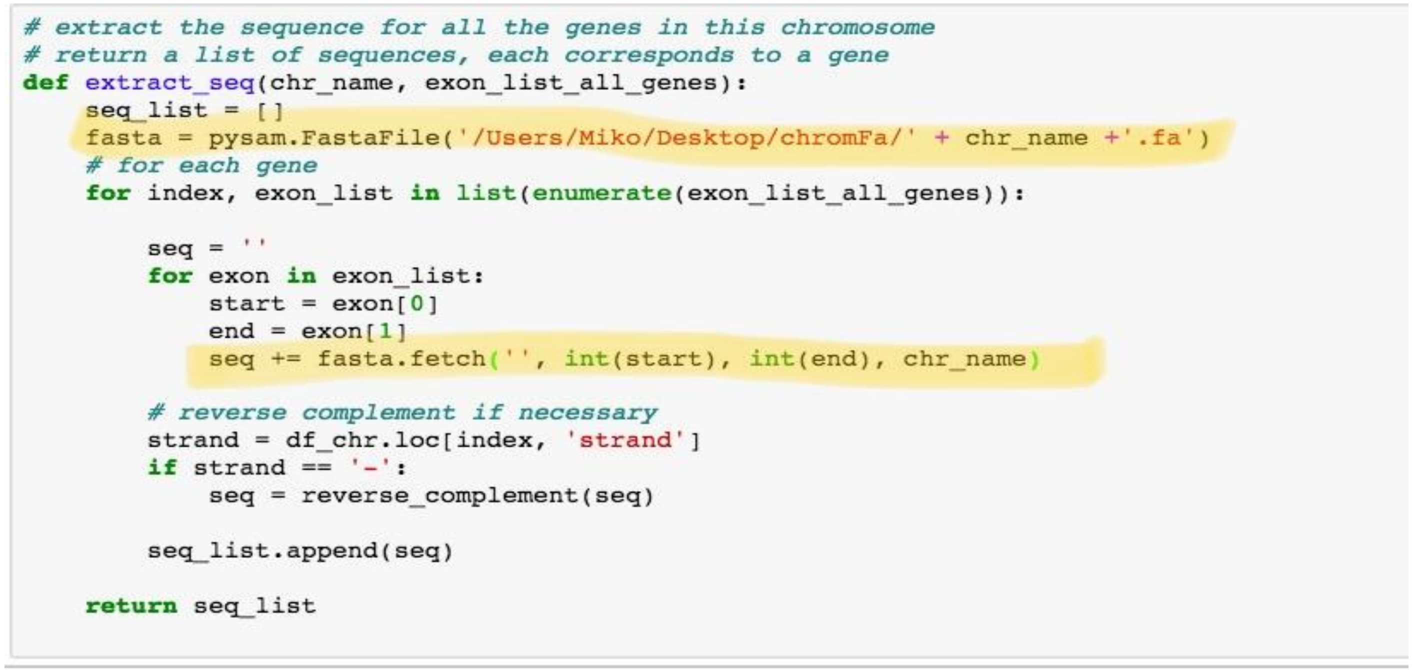
STEP 2: Get all the <u>sequences</u> of translated exons

- Using Pysam, to get sequence from fasta file
- Reverse complement the sequences on negative (-) strand



```
 'TGTGTTGCATACTTTCTAAGGCGGCGGCTGCAGCAGCGGCTCCATCCAGCCCGTCAGCTCCTCCTGCAAGGCATGGCTGGCTACCTGAGTGAATCGGACTTTGTGATGGTGGAGG
AGGGCTTCAGTACCCGAGACCTGCTGAAGGAACTCACTCTGGGGGCCTCACAGGCCACCACGGACGAGGTAGCTGCCTTCTTCGTGGCTGACCTGGGTGCCATAGTGAGGAAGCACT
TTTGCTTTCTGAAGTGCCTGCCACGAGTCCGGCCCTTTTATGCTGTCAAGTGCAACAGCAGCCCAGGTGTGCTGAAGGTTCTGGCCCAGCTGGGGCTGGGCTTTAGCTGTGCCAACA
AGGCAGAGATGGAGTTGGTCCAGCATATTGGAATCCCTGCCAGTAAGATCATCTGCGCCAACCCCTGTAAGCAAATTGCACAGATCAAATATGCTGCCAAGCATGGGATCCAGCTGC
TGAGCTTTGACAATGAGATGGAGCTGGCAAAGGTGGTAAAGAGCCACCCCAGTGCCAAGATGGTTCTGTGCATTGCTACCGATGACTCCCACTCCCTGAGCTGCCTGAGCCTAAAGT
TTGGAGTGTCACTGAAATCCTGCAGACACCTGCTTGAAAATGCGAAGAAGCACCATGTGGAGGTGGTGGGTGTGAGTTTTCACATTGGCAGTGGCTGTCCTGACCCTCAGGCCTATG
CTCAGTCCATCGCAGACGCCCGGCTCGTGTTTGAAATGGGCACCGAGCTGGGTCACAAGATGCACGTTCTGGACCTTGGTGGTGGCTTCCCTGGCACAGAAGGGGCCAAAGTGAGAT
TTGAAGAGATTGCTTCCGTGATCAACTCAGCCTTGGACCTGTACTTCCCAGAGGGCTGTGGCGTGGACATCTTTGCTGAGCTGGGGCGCTACTACGTGACCTCGGCCTTCACTGTGG
CAGTCAGCATCATTGCCAAGAAGGAGGTTCTGCTAGACCAGCCTGGCAGGGAGGAGGAAAATGGTTCCACCTCCAAGACCATCGTGTACCACCTTGATGAGGGCGTGTATGGGATCT
TCAACTCAGTCCTGTTTGACAACATCTGCCCTACCCCCATCCTGCAGAAGAAACCATCCACGGAGCAGCCCCTGTACAGCAGCAGCCTGTGGGGCCCGGCGGTTGATGGCTGTGATT
GCGTGGCTGAGGGCCTGTGGCTGCCGCAACTACACGTAGGGGACTGGCTGGTCTTTGACAACATGGGCGCCTACACTGTGGGCATGGGTTCCCCCTTTTGGGGGACCCAGGCCTGCC
ACATCACCTATGCCATGTCCCGGGTGGCCTGGGAAGCGCTGCGAAGGCAGCTGATGGCTGCAGAACAGGAGGATGACGTGGAGGGTGTGTGCAAGCCTCTGTCCTGCGGCTGGGAGA
TCACAGACACCCTGTGCGTGGGCCCTGTCTTCACCCCAGCGAGCATCATGTGAGTGGGCCTCGTTCCCCCGGAGAATCCCAGCGGGGCCTCAGAGATGCATCTGGGAGAGGTGGGG
AAGATGGCAGGCAAGGGTACCCTTGGCCAGGACTCTGGTGCCCACCCTGCCACCCCCGCGCTCCACCTGCAGTGTTTCTGCCCTGTAAATAGGACCAGTCTTACACTCGCTGTAGTT
CAAGTATGCAACATAAATCCTGTTCCTTCCA',
```

Extract sequences using Pysam package



```python
# extract the sequence for all the genes in this chromosome
# return a list of sequences, each corresponds to a gene
def extract_seq(chr_name, exon_list_all_genes):
    seq_list = []
    fasta = pysam.FastaFile('/Users/Miko/Desktop/chromFa/' + chr_name +'.fa')
    # for each gene
    for index, exon_list in list(enumerate(exon_list_all_genes)):

        seq = ''
        for exon in exon_list:
            start = exon[0]
            end = exon[1]
            seq += fasta.fetch('', int(start), int(end), chr_name)

        # reverse complement if necessary
        strand = df_chr.loc[index, 'strand']
        if strand == '-':
            seq = reverse_complement(seq)

        seq_list.append(seq)

    return seq_list
```

Methods

STEP 3: Suppose there is an insertion or deletion of 1 or 2 nucleotide at a position in an exon, can we rescue the mutation using alternative splicing (exons skipping)?

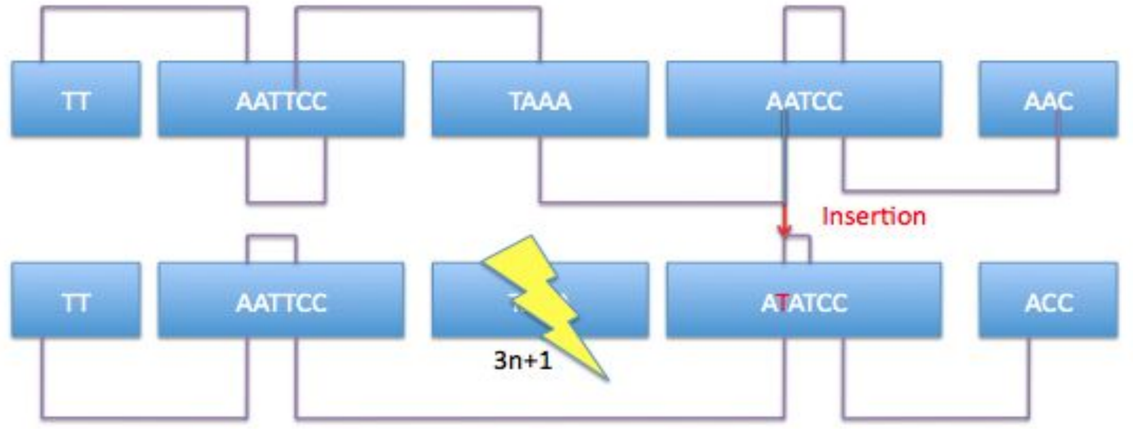
- Four cases

Four cases discussion

--delete the previous or next exon with the base pair of length...

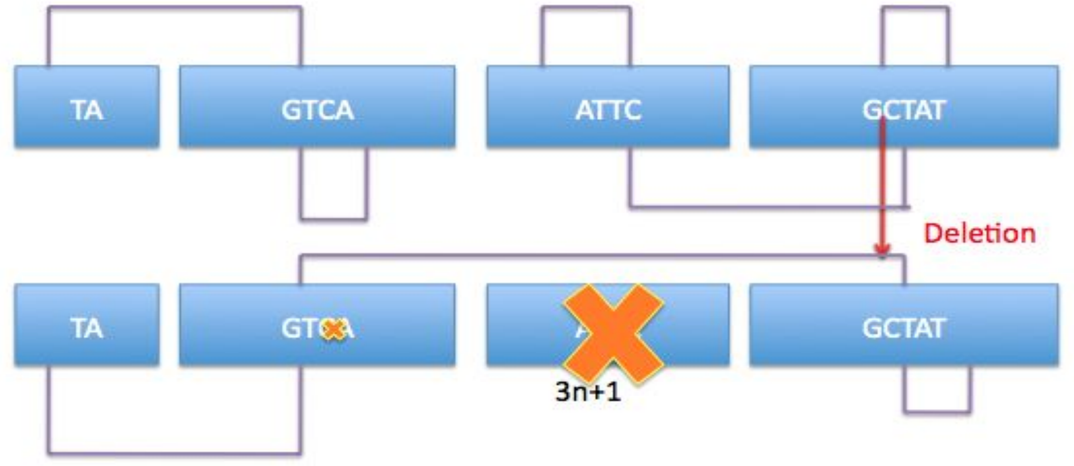
- Insert 1 nucleotide: 3n+1
- Insert 2 nucleotide: 3n+2
- Delete 1 nucleotide: 3n+2
- Delete 2 nucleotide: 3n+1

Example illustration 1: Insert 1 nucleotide



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

Example illustration 2: Delete 2 nucleotides



The original frame: TAG, TCA, ATT, CGC, TAT
The Frame after splicing : TAG, TGC, TAT

General idea

- Suppose there is an insertion of one nucleotide
- Check the length of previous and/or next exon
- Check if stop codon would be created by the splicing
- RESCUE == right length of a exon to skip & no stop codon created

Summary

- **Input**: the sequences of exons
- **Goal**: Try to rescue the frameshift mutation using alternative splicing/ exon skipping
- **Output**: lists for each exon, indicate which frameshift mutations can be rescued

Gene	Exon	Position	frameshift	Exon skipped	Comments
CHRNE	6	45-72	+2	5	
ADC	4	33549670	+2	5	
ADC	6	33558968	+1	7	
NECAP2	7	16785405	+1	6	
...	...	...	...	...	

Thank you!