RNA SEQUENCE PARSER

Mumuksha Pant

1. Introduction to Problem

RNA (Ribonucleic Acid) is a crucial molecule in biological systems, playing various roles in coding, decoding, regulation, and expression of genes. RNA sequences are composed of four types of nucleotides: Adenine (A), Cytosine (C), Guanine (G), and Uracil(U). These sequences can be represented as strings consisting of the letters A, C, G, and U.

Ambiguity codes are used in DNA and RNA sequencing to represent situations where a specific nucleotide at a particular position cannot be determined with certainty. We have built a Python parser for RNA sequences handling both standard nucleotide (A, C, G , T) and nucleotide with ambiguity codes.

Part I of the problem focusses on creating an RNA Parser to identify the nucleotide composition.

1.1 Sequence Validation

the function is_valid_rna validates if a given string is a valid RNA sequence (contains only A, C, G, U). The function returns True for valid sequences and False for invalid ones.

Approach

We are using Regular Expression (^[ACGU]+$) to validate the sequence contains

(A, G, C , U) from start to end ignoring the case sensitivity. The function returns True for valid sequences and False for invalid ones.

1.2 Nucleotide Count

the function nucleotide_count counts the occurrence of each nucleotide (A, C, G, U) in a given RNA sequence and returns a dictionary with nucleotides as keys and their counts as value.

Approach

We use a dictionary to keep a count of A, G, C , U and iteratively iterate the RNA sequence while incrementing the count for the nucleotides.

1.3 Finding Motifs

the function identifies and returns all occurrences of a given motif (subsequence) within the RNA sequence.

Approach

we use the regular expressions finditer( ) function to search for all occurence of the specified pattern, ie, motif. The position is stored in a list called “position” . The RNA sequence is iterated & start index are appended in the “position” till no more motif are found.

1.4 Sequence Complementarity

the function 'complementary_sequence' generates the complementary sequence of a given RNA sequence by swapping pairs.

Approach

we iterate over the RNA sequence and for every pair we swap the two nucleotides then append in into the resulting list “complementary_sequence_list” . After the swapping is complete we return the list.

1.5 GC Content Calculation

the function 'gc_content’ calculates the GC content (percentage of nucleotides G and C) in the RNA sequence, which is significant in determining the stability of the molecule.

Approach

we are using regular expressions findall( ) to find the occurence of ‘G’ and ‘C’ then calculate its length . The gc_content can be found by using the formula

$$GCcontent = \frac{GCcounts}{lengthofthesequence}$$

Part II of the problem focusses on creating the RNA Parser acknowledging the presence of variability or ambiguity in nucleotide sequences. Each ambiguity code represents a different combination of the standard nucleotides.

2.1 Advanced Sequence Validation

the function “is_valid_rna_modified” is the modified version of the ‘is_valid_rna’ function which checks for commonly used ambiguity codes in RNA sequences (e.g., N for any nucleotide, R for A or G) and validate accordingly.

Approach

the function uses regular expression search( ) function to match the given sequence of RNA to the pattern *r' ^[AGCURYSWKMBDHVN]+$ '* .

2.2 Regex-based Motif Search with Ambiguities

the function "find_motifs_modified" adapts the 'find_motifs' function to accept motifs with ambiguity codes and identify potential matches in the sequence.

Approach

The approach is similar to Finding Motifs (1.3) where we use finditer( ) to search for the pattern. We create a dictionary of ambiguity codes corresponding to a set of nucleotides. The function then iterates over each character in the motif. If the character is found in the ambiguity_codes, it appends the corresponding regex pattern to the res list; otherwise, it appends the character itself.

2.3 Sequence Fragmentation and Analysis

the function called 'fragment_and_analyze' fragments the RNA sequence into smaller segments of a specified length and performs a detailed analysis on each fragment including '*is_valid_rna*', '*gc_content*' and '*complementary_sequence*' .

Approach

we are utilizing the "*is_valid_rna_modified*" , "*gc_content_modified*" and "*complementary_sequence_modified*" functions to process each fragment of the specified fragment length of the provided RNA sequence.

Challenges Faced

1) Handling Ambiguity Codes

one of the challenges were modifying the functions to handle ambiguity codes. Example modifying gc_content calculation & using it in sequence fragmentation function. Initially the results for gc_content were wrong for a fragment of RNA sequence “GCRYSN” because our original gc_content function did not consider ’S’ . ’S’ is 100% GC content (C , G)

2) Fragmentation and Analysis

This function took the most time to develop because some changes were required in its helper functions .

Possible Improvements

1. **Edge cases for testing**
   for a proper exhaustive testing, we can add more edge cases to check for better error handling. We can add better error messages .

2. **Optimizing the function Motif Search with Ambiguities**
   if the RNA sequence is very large, the function could take up a lot of time.