Journal of Computational Biology (1) 2021 01-09 Submitted 09/13; Published 09/13

First CSCI-551 Assignment, Supplemented by Sung (2009) and Gusfield (1997)

Joshua Ryan Steenson joshua.ryan.steenson@gmail.com
Jordan Dood jordan.dood7@gmail.com
Susan McCartney susanmccartney12@gmail.com

Editor: Joshua Ryan Steenson, Jordan Dood, and Susan McCartney

First Problem

For the textbook problem encompassed by this first homework problem, the goal is to extract the **5′ Untranslated Region (UTR),** the **3′ Untranslated Region (UTR),** and the protein sequence for the given **mRNA** sequence (Sung (2009)).

$$\mathbf{5'\text{–}ACTTGTCATGGTAACTCCGTCGTACCAGTAGGTCATG\text{–}3'}$$

$$\mathbf{3'\text{–}TGAACAGTACCATTGAGGCAGCATGGTCATCCAGTAC\text{–}5'}$$

First and foremost, the given **mRNA** sequence is presented in the language of **Deoxyribonucleic Acid (DNA)**. The sequence must be converted to the language of **Ribonucleic Acid (RNA),** or more accurately, the language of **Messenger RNA (mRNA)**.

$$\mathbf{5'\text{–}ACUUCUCAUGGUAACUCCGUCGUACCAGUAGGUCAUG\text{–}3'}$$

The coding region of this **mRNA** sequence begins with a start codon. In the genetic code, the start codon **AUG** encodes **Methionine (Met)** (Sung (2009)). In our **mRNA** sequence, the codon **AUG** exists. Everything preceding the start codon is considered to be the **5′ UTR**.

5′ UTR	**5′–ACUUCUC**

The coding region begins with **AUG** and ends with a stop codon, which in the context of this **mRNA** sequence is **UAG,** a stop codon that terminates the given coding region (**Sung** (**2009**)). Given this range, the following protein sequence unfolds, so to speak.

AUG–GUA–ACU–CCG–UCG–UAC–CAG–UAG

In our coding region, the codon **AUG** represents **Methionine (Met),** the codon **GUA** represents **Valine (Val),** the codon **ACU** represents **Threonine (Thr),** the codon **CCG** represents **Proline (Pro),** the codon **UCG** represents **Serine (Ser),** the codon **UAC** represents **Tyrosine (Tyr),** the codon **CAG** represents **Glutamine (Gln),** and the codon **UAG** represents the stop codon that terminates the coding region (**Sung** (**2009**)).

Met–Val–Thr–Pro–Ser–Tyr–Gln–Ter

After the coding region, we have the **3′ UTR**. What is interesting about the **3′ UTR** for this **mRNA** sequence is that it has a start codon contained within it. However, it does not lead to another region because there is no stop codon. Ultimately, the **3′ UTR** is presented as it is because of the enormous variety of **mRNA** sequences. Also, more importantly, the objective of this exercise was, given a finite sequence that is provided as it is presented, to identify the required components. Such an objective has been accomplished as best as possible. The structure of a gene also includes a regulatory region preceding the **5′ UTR,** but more information would be required to characterized such a region for this sequence, or for any given sequence (**Sung** (**2009**)).

3′ UTR	**GUCAUG–3′**

Second Problem

Exact Pattern Matching. Given a very long text string, denoted **T,** and a short pattern string denoted **P,** find all occurrences of **P** in **T,** such that the length of **T** is equal to **n** and the length of **P** is equal to **m** (Gusfield (1997)).

Because there are $\boldsymbol{n}$ iterations for the ***Z*-Algorithm,** because the number of matches is less than or equal to $\boldsymbol{n}$ given that each match increases by the value of $\boldsymbol{r}$**,** and because the number of mismatches is less than or equal to $\boldsymbol{n}$ such that the number of mismatches is less than or equal to one per iteration, the overall running time of the ***Z*-Algorithm** is calculated as $\mathcal{O}(\mathbf{n})$ (Gusfield (1997)).

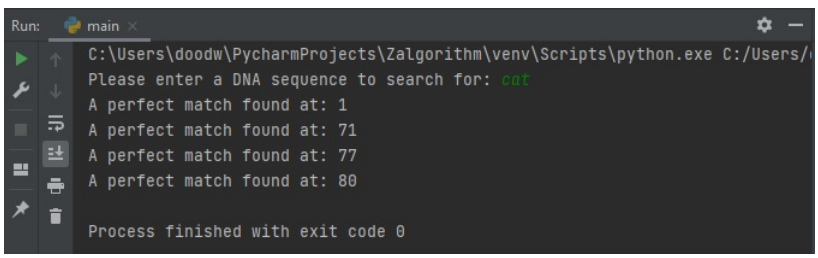

Figure 1: Command Line Run

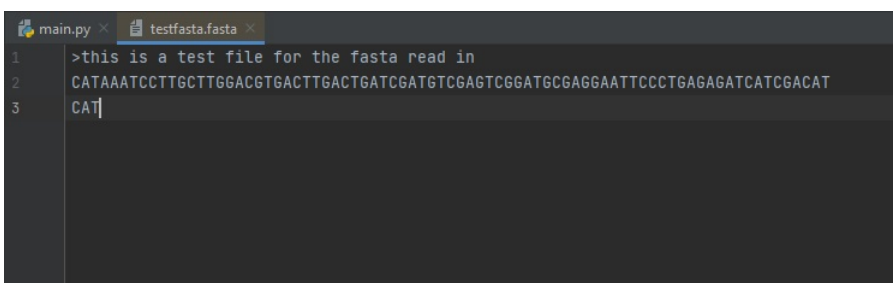

Figure 2: File to Check Algorithm Against

```
Initialize l, r = 0 ;                    /* For the main loop, compute Z_k */
for k = 2 to n do
    if k > r then
        Find Z_k by comparing S[k...n] and S[1...n];
        if Z_k > 0 then
            Set r = k + Z_k − 1 and set l = k;
        end
    else
        Let k' = k − l + 1 and |β| = r − k + 1;
        if Z_k' < |β| then
            Z_k = Z_k' ;                                  /* l, r Unchanged */
        else
            Compare S[|β| + 1, ...n] with S[r + 1, ...n] until a mismatch
              occurs at position q ≥ r + 1;
            Set Z_k = q − k, r = q − 1, and l = k;
        end
        if no mismatches occur then
            q = n + 1;
        end
    end
end
```

Algorithm 1: Z-Algorithm (Gusfield (1997))

References

D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.

W. Sung. *Algorithms in Bioinformatics: A Practical Introduction.* CRC Press, 2009.