Searching Open Reading Frame in a DNA Sequence using Dynamic Programming

by

The 3rd International Conference On Science
Journal of Physics: Conference Series **1341** (2019) 092010 doi:10.1088/1742-6596/1341/9/092010

Searching Open Reading Frame in a DNA Sequence using Dynamic Programming

Y A Yunus[1], A Lawi[1], S A Thamrin[2] and W Musu[3]

[1]*Computer Science, Fac. Mathematics and Natural Science, Hasanuddin University*
[2]*Statistics, Faculty of Mathematics and Natural Science, Hasanuddin University*
[3]*Department of Informatic Engineering, STMIK Dipanegara Makassar*

E-mail: armin@unhas.ac.id, yayuaprilika@gmail.com, tuti@unhas.ac.id, wilem.musu@dipanegara.ac.id

Abstract. Aside from being a carrier of information on the nature of the organism or cell, DNA also plays an important role in the formation of amino acids. Before coding nitrogenous base sequences in DNA to amino acids, there is a search process for open reading frames (ORFs) which will produce parts of DNA sequences that have the potential to become amino acids. One of the problems that arise in ORF search is that there is still a lack of computational programs to get ORF. Therefore, this study tries to make ORF search programs by implementing dynamic programming that is widely used in the field of bioinformatics using python applications. In addition, the importance of displaying all the possibilities of ORF in a DNA sequence is that it can help research in the field of genetic analysis.

1. Introduction

Genes are information carrier substances that influence the characteristics of an organism that is passed on from generation to generation. Genes consist of deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). DNA is a polymer consisting of four different types of monomers called nucleotides [1].

Aside from being a carrier of information on the nature of the organism or cell, DNA also plays an important role in the process of protein formation. The process of changing each protein is composed of a number of amino acids in a certain sequence and each amino acid formation is encoded by the sequence of nitrogen bases contained in DNA.

Before coding the nitrogen base sequence in DNA, there is a search process for open reading frames (ORF) which will produce parts of DNA sequences that have the potential to become amino acids. There are several websites that provide the ORF Finder page but do not display the overall ORF results on the DNA sequence that has been encased.

One of the problems that arise in ORF search is that there is still a lack of computational programs to get ORF. Therefore, this study tries to make ORF search programs by implementing dynamic programming that is widely used in the field of bioinformatics. In addition, the importance of displaying all the possibilities of ORF in a DNA sequence is that it can help research in the field of genetic analysis.

This paper is organized as follows. We explain the introduction in the first part, followed by the material and methods. In this second part, we explain how to apply dynamic programming methods to ORF searches. In the next section, we present the results. Conclusions and future works are presented further in the last section.

2. Materials and Methods

2.1. Study Area

Nucleotides are simple subunits that make up DNA consisting of sugar (deoxyribose) with phosphate groups attached to it, and nitrogen bases, in the form of adenine, guanine, cytosine and thymine. So nucleotides are labeled A, G, C, and T [2].

Open Reading Frame (ORF) is a triplet nucleotide sequence that is read as a codon that determines amino acids, one DNA strand has three possible reading frames. The length of ORF can indicate the coding region of the candidate protein in the DNA sequence [3]. The characteristics of an ORF are as follows:

1. The initiation codon or start codon.
   A start codon in DNA begins the translation of the first amino acid in a polypeptide chain. The first three bases of the coding sequence of the mRNA to be translated into proteins, are where the initiation codons are. ATG is the most common start codon in DNA and a code for methionine amino acids (Met) in eukaryotes and formyl methionine (fMet) in prokaryotes. During protein synthesis, ATG is converted into AUG in mRNA which is then recognized by tRNA as a start codon with the help of several initiation factors and begins the translation of mRNA [4].
2. There is a series of nucleotide base sequences which are found between the start and stop codon positions in ORF.
3. Termination codon or Stop codon (TAA, TAG, TGA on DNA).
   A stop codon is actually three nucleotide bases in the messenger RNA that mark the termination of the translation process. Most of the codons in mRNA are related to the addition of specific amino acids to the growing protein chain, in a certain order. The formation of an amino acid chain will stop at this point [5].

2.2. Data

We applied this method using publicly available fasta NC_001477 data at (https://www.ncbi.nlm.nih.gov/nuccore/NC_001477). This fasta data contains 10,735 DNA nucleotide sequences from type 1 Dengue virus [6] and we use the python program language.

2.3. Dynamic Programming

Dynamic Programming is like a divide-and-conquer method that solves problems by combining solutions to sub-problems. The divide-and-conquer algorithm describes problems into sub-problems recursively, sub-problems, and then combines each solution to solve the original problem. However, dynamic programming applies when sub-problems overlap (conditions where sub-problems are divided into the same sub-problems). Dynamic programming algorithms solve each sub-problem only once and then store the answer in a place, by avoiding the work of recomputing the answer each time completing each sub-problem [7]. In addition to the recursive concept, dynamic programming is also known as memoisasi or recording which is a description of optimization techniques by storing calculated results in a memo or in other words stored in a variable then the calculated results will be recalled when they will be used in the next process [8], this memoization technique from dynamic programming will be used later.

2.4. Biopython

Biopython is a set of libraries that exist in python software for computing in the field of bioinformatics. Python is a programming language that emphasizes the readability of code so that it is easier to understand the syntax. This language first appeared in 1991, designed by a man named Guido van Rossum who until now is still being developed by the Python Software Foundation. The biopython website provides online resources for modules, scripts, and web links for bioinformatics use and research [9].

Biological sequences or sequences can be said to be the central objects in Bioinformatics, and in this library there are several mechanisms to handle sequences, namely package Sequence Objects, Sequence Annotation Objects and Sequence Input / Output and so on. Input / Output Package Sequence that provides a simple interface for the input output process for input data circuits with various data formats, one of which is fasta which is the data format in this study.

3. Discussion and Result

The stage in this study begins with the process of inputting DNA sequence data and codons, which then search for codon start and stop codon by using dynamic programming.

```
from Bio import SeqIO
data=SeqIO.read("dengue fasta","fasta")
seq = str(data.seq)
startcodon = ["a","t","g"]
stopcodon = [["t","a","g"],["t","g","a"],["t","a","a"]]
```

Figure 1 Input Data

The data input process in Figure 1 involves using the SeqIO package from the biopython library. The SeqIO.read command functions to read the nucleotide circuit data from the file "dengue fasta" stored to the "seq" variable with the str (string) data type and startcodon and stopcodon as the genetic code that marks the open reading frame stored in the list.

The process of applying the dynamic programming method in this study is broken down into several stages. The first stage is to find the start codon and stop codon, the results of this step are then used to select each ORF which allows it to become an amino acid. At this stage, each start and stop position in the nucleotide sequence will be saved in the new list.

3.1. Start Codon

In Figure 2 a loop is performed along the sequence data of seq and given a variable a value of 0, b has an initial value equal to the index j value of 0. Next conditioning is done if the contents of the index seq [b] = startcodon [a] then the next coding will be done with the values of a and b increase by 1 and j = 0, as in Figure 3. If all three conditions are met at a given value j, then each start codon found in the index number seq will be stored in the list daftarstart.append[], this process is shown in Figure4. The function of this append is to add items to the new list from behind. The index number found is then recognized as the start position of the codon. daftarstart.append[] this will later be recalled when determining ORF.

```
for j in length seq:
    a=0
    b=j
    if seq[b]= startcodon[a]:
      a+=1
      b+=1
      if seq[b]=startcodon[a]:
        a+=1
        b+=1
        if seq[b]=startcodon[a]:
          daftarstart.append(j)
```

Figure 2 Pseudocode Start Codon

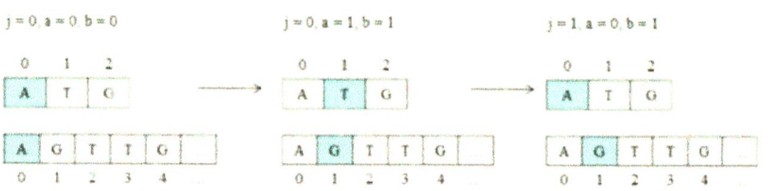

Figure 3 Loop Illustrations

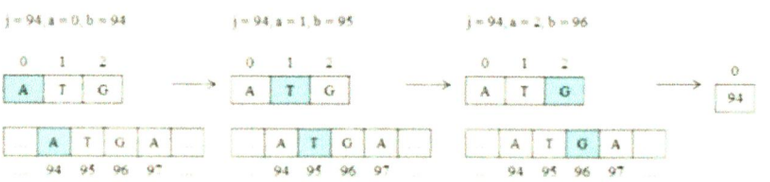

Figure 4 Startcodon conditioning

3.2. Stop Codon

Stop codon is marked with "TAG", "TGA", and "TAA". Then multi-dimensional lists are used to have the list concept in the list. To process a 2-dimensional list, it is necessary to use nesting loops. The process of working on the stop codon search is similar to startcodon search if seen in Figure 5 but there are conditions that use 2 dimensions in this process.

```
for i in length stopcodon :
  for j in length seq :
    a=0
    b=j
    if seq[b]=stopcodon[i][a]:
      a+=1
      b+=1
      if seq[b]=stopcodon[i][a]:
        a+=1
        b+=1
        if seq[b]=stopcodon[i][a]:
          daftarstop.append(j)
```

Figure 5 Pseudocode Stop Codon

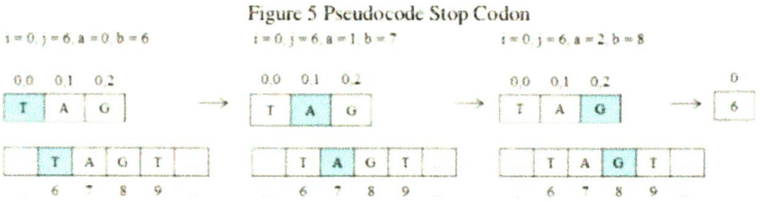

Figure 6 Stopcodon conditioning

In Figure 6 when the value i = 0, j = 6, a = 0 and b = 6, the contents of index seq[6] are the same as the contents of index stopcodon [0] [0], so the first conditioning is fulfilled, as well as subsequent conditioning. If one of the conditioning is not fulfilled, the looping j in range of the stop

stop will increase 1. If the loop j has been completed and the loop i increases by 1 to meet the range of the range from the stop list.

3.3. Determination of Open Reading Frame

At this stage, the memoization process occurs, namely the required subproblems daftarstart and daftarstop are completed first and then used to build solutions to larger problems. Based on Figure 7 the loop is nested using a while against the list that was created in Figure 2 start codon and in Figure 5 stop codon. The first loop is done as long as index i is smaller than the daftarstart list length of 292 and the second loop is done as long as the index j is smaller than the length of the daftarstop list which is 452. Figure 8 shows part of the list daftarstart and daftarstop when the value i = 0 and j = 0, the variable a = j is defined so that the value of a on the first loop is 0. The initial conditioning in Figure 7 is not fulfilled when the value in the daftarstop <= daftarstart so that the value of j increases one as in Figure 8.

```
i=0 ; j=0
while i<len(daftarstart):
  b=0
  while j<len(daftarstop):
    a=j
    if daftarstop[j]>daftarstart[i]
                |
    else:
      j+=1
  i+=1
  j=a
```

Figure 7 Pseudocode ORF

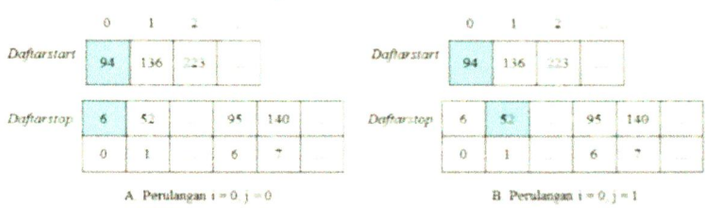

Figure 8 Conditioning is not fulfilled

Figure 9 shows the looping process when the daftarstop > daftarstart requirement is not met. In Figure 9 (a), the nucleotide startcodon in the index start list i = 0 contains 94 and in the daftarstop index j = 2 contains 57 so that the conditioning is not met. Thus, repetition increases by one to j = 3 containing 63 (Figure 9 (b)) and does not meet conditioning 'daftarsdtop[j]> daftarstart [i] to j = 4 in Figure 9 (c). However, Figure 9 (d) the startcodon nucleotide in the index daftarstart i = 0 contains 94 and in the index daftarstop j = 6 contains 95 so that the conditioning of Figure 7 daftarstop [j]> daftarstart [i] is fulfilled. Previously described in the literature review ORF requires the presence of nucleotides other than startcodon and stopcodon so that the next conditioning is made where the contents of the index daftarstop[j] - daftarstart [i]> 3 (Figure 10).

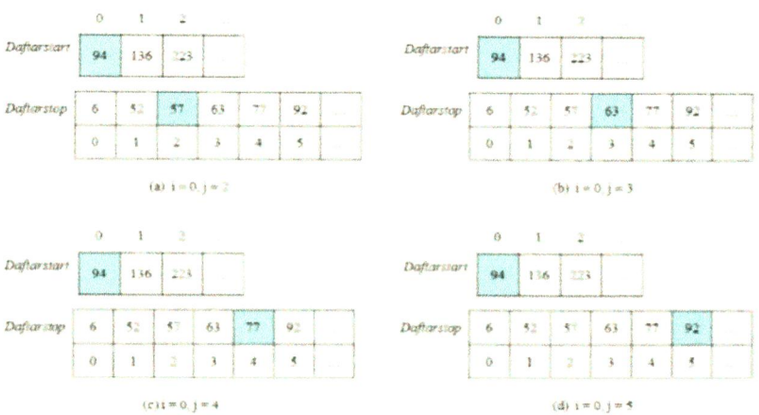

Figure 9 Repetition process

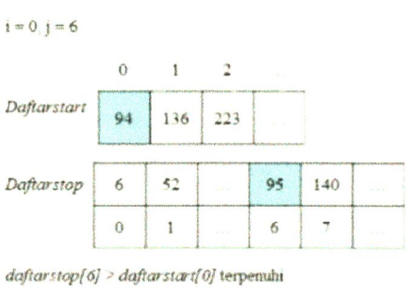

Figure 10 i = 0 and j=6

Stopcodon consists of 3 nucleotides, on the index stop list j = 6 is 95, from position 95 = T, 96 = G, and 97 = A is stopcodon (TGA). daftarstart at index i = 0 is 94 from startcodon position where 94 = A, 95 = T, 96 = G. If ORF is arranged from index i = 0 and j = 6 as in Figure 11 where start codon and stop codon use the same nucleotide and break the ORF rules. Therefore, the contents of the "stop list" index at least have a position difference greater than 3

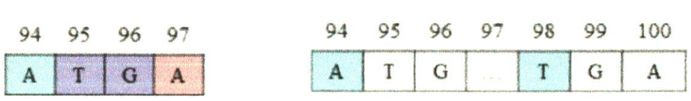

Figure 11 Second Condition

Because at index i = 0 and j = 6 does not fulfill to be ORF, index i = 0 and j increases one to j = 7 which then fulfills the requirement daftarstop > daftarstop, then the next conditioning is daftarstop[7] - daftarstart[0] "= 140 - 94 = 46 where 46 > 3 is fulfilled

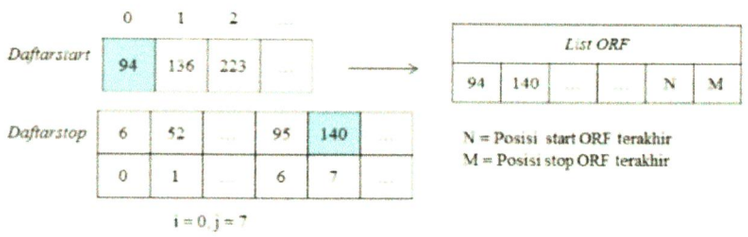

Figure 12 Conditioning is fulfilled

In Figure 12 shows the fulfilled conditioning, the values of daftarstart[i] and daftarstop[j] are input to the ORF list, where 94 marks the first start ORF position entered in the first index ORF list and 140 marks the first stop ORF position entered into the second index. The process of input to this ORF list will continue until the last ORF is found from the nucleotide sequence used. From this list, print an output of 288 orf.

4. Conclusion

In this paper, inputting DNA nucleotide data in python involves the Biopython library using the *SeqIO* package, which serves to read circuit files with various data formats. Dynamic programming methods are applied to solve search problems into the start and stop codon position search stages which are stored in the new list and then processed again when determining ORF. In addition, the startcodon index is smaller than the stopcodon index in order to form each ORF series and find out the length of each ORF by reducing the index value of daftarstop with a daftarstart so that 288 ORF is obtained.

References

[1] Neil A. Campbell and Jane Reece. *Campbell Biology 11th Edition*. New York: Pearson Higher Education, 2003.

[2] Bruce Albert and Alexander Johson. *Molecular Biology of The Cell Sixth Edition*. New York: Garland Science, 2014.

[3] Nature com. nature.com. [Online]. https://www.nature.com/subjects/open-reading-frames Accessed on November 16, 2018.

[4] Biology Wise Staff. (2018) Biology Wise. [Online]. https://biologywise.com/start-codon Accessed on September 29, 2018.

[5] BiologyWiseStaff. (2018) Biology Wise. [Online]. https://biologywise.com/stop-codon Accessed on September 29, 2018.

[6] ncbi.com. NCBI. [Online]. https://www.ncbi.nlm.nih.gov/nuccore/NC_001477 Accessed on Oktober, 2018.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction To Algorithms Third Edition*. London: The MIT Press, 2009.

[8] rahmadya. (2018, November) rahmadya. [Online]. https://rahmadya.com/2018/11/12/pemrograman-dinamis-konsep-memoization/ Accessed on January 19, 2019.

[9] Jeff Chang, Brad Chapman, Iddo Friedberg, and Thomas Hamelryck. (2018) Biopython. [Online]. http://biopython.org/DIST/docs/tutorial/Tutorial.html Accessed on December, 2018.

Searching Open Reading Frame in a DNA Sequence using Dynamic Programming

ORIGINALITY REPORT

%10	%6	%3	%8
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Laredo Community College Student Paper	%1
2	koofers.com Internet Source	%1
3	Submitted to Universitas Siswa Bangsa Internasional Student Paper	%1
4	Submitted to Florida Virtual School Student Paper	%1
5	Nicolae Ţăndăreanu. "Distinguished representatives for equivalent labelled stratified graphs and applications", Discrete Applied Mathematics, 2004 Publication	%1
6	Submitted to SASTRA University Student Paper	%1
7	www.news-medical.net Internet Source	%1

8	biopython.org Internet Source	%1
9	Submitted to Oldham Sixth Form College Student Paper	<%1
10	Submitted to University of Nebraska at Omaha Student Paper	<%1
11	lists.pirateweb.net Internet Source	<%1
12	Submitted to Luton Sixth Form College, Bedfordshire Student Paper	<%1
13	www.scribd.com Internet Source	<%1

EXCLUDE QUOTES ON

EXCLUDE BIBLIOGRAPHY ON

EXCLUDE MATCHES < 5 WORDS