RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Program: Biomedical Engineering

Course Number	**BME 808**
Course Title	**Computations in Genetic Engineering**
Semester/Year	Winter 2021
Instructor	P. Siddavaatam

Lab No.	**4**

Report Title	**Gene Annotation by sequence**

Section No.	2
Group No.	n/a
Submission Date	04/10/2021
Due Date	04/10/2021

Name	Student ID	Signature*
Andrew Mullen	500 787 631	

(Note: remove the first 4 digits from your student ID)

* By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your Academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.

BME 808 Lab 4
Gene Annotation by Sequence

1.

```
def findPattern(pattern, seq, start, stop, inc, thresh) :
    lengthPat = len(pattern)
    lengthSeq = len(seq)
    start = max(0,start)
    stop = min(stop, lengthSeq-lengthPat)

    i = start

    while i<=stop :
        j = 0
        count = 0
        for k in range(lengthPat) :
            if seq[i+j] == pattern[k] :
                count += 1
            j+=1
        if count >= thresh:
            return i

        i = i+inc

    return -1
```

```
# Find first ATG position
findPattern('ATG', seq, 0, len(seq), 1, 3)
```

```
37
```

2.

```
# Look for gene --> Requirements ATG and Stop Codon
SHORT = 99
i = 0
zz = 0
while i<len(seq)-SHORT :
    # Find first ATG index
                    # pattern, sequence, start, stop, inc, thresh
    atg = findPattern('ATG', seq, i, len(seq)-SHORT, 1, 3)


    if atg == -1 :
        print('No more ATGs found')
        break

    # Find the first stop index after the ATG index
    stops = []
    tag = findPattern('TGA', seq, atg+3, len(seq)-SHORT, 3, 3)
    tta = findPattern('TAA', seq, atg+3, len(seq)-SHORT, 3, 3)
    tga = findPattern('TGA', seq, atg+3, len(seq)-SHORT, 3, 3)

    # Store the lcation of the stops
    if tag != -1 :
        stops.append(tag)
    if tta != -1 :
        stops.append(tta)
    if tga != -1 :
        stops.append(tga)

    if len(stops) == 0 :
        print('No more stops for that ATG')
        i = 100000
        continue

    i = atg + 1

    minStop = min(stops)
    if minStop-atg < SHORT :
        #print('Too short')
        continue
    i = minStop + 1
    zz+=1
    print('\nORF #', zz, end =' ')
    print('ATG', atg, end = ' ')
    #print('---------')
    print('Stop at:', minStop)
```

ORF # 1 ATG 37 Stop at: 592

ORF # 2 ATG 745 Stop at: 1438

ORF # 3 ATG 1477 Stop at: 2569

ORF # 4 ATG 2575 Stop at: 2791

ORF # 5 ATG 2870 Stop at: 3797

ORF # 6 ATG 3897 Stop at: 4776

ORF # 7 ATG 4784 Stop at: 5390

ORF # 8 ATG 5820 Stop at: 6363

ORF # 9 ATG 6482 Stop at: 6614

ORF # 10 ATG 7536 Stop at: 7731

ORF # 11 ATG 7962 Stop at: 8697

ORF # 12 ATG 8704 Stop at: 8833

ORF # 13 ATG 9132 Stop at: 9258

ORF # 14 ATG 9785 Stop at: 9923

ORF # 15 ATG 10000 Stop at: 10180

ORF # 16 ATG 10347 Stop at: 11058

ORF # 17 ATG 11110 Stop at: 11368

ORF # 18 ATG 11468 Stop at: 11588

ORF # 19 ATG 11671 Stop at: 12076

ORF # 20 ATG 12233 Stop at: 12905

ORF # 21 ATG 12934 Stop at: 13177

ORF # 22 ATG 13282 Stop at: 14149

ORF # 23 ATG 14183 Stop at: 14864

ORF # 24 ATG 14899 Stop at: 15805

3.

```
SHORT = 99
i = 0
zz = 0
while i<len(seq)-SHORT :
    # Find first ATG index
                    # pattern, sequence, start, stop, inc, thresh
    atg = findPattern('ATG', seq, i, len(seq)-SHORT, 1, 3)


    if atg == -1 :
        print('No more ATGs found')
        break

    # Find the first stop index after the ATG index
    stops = []
    tag = findPattern('TGA', seq, atg+3, len(seq)-SHORT, 3, 3)
    tta = findPattern('TAA', seq, atg+3, len(seq)-SHORT, 3, 3)
    tga = findPattern('TGA', seq, atg+3, len(seq)-SHORT, 3, 3)

    # Store the lcation of the stops
    if tag != -1 :
        stops.append(tag)
    if tta != -1 :
        stops.append(tta)
    if tga != -1 :
        stops.append(tga)

    if len(stops) == 0 :
        print('No more stops for that ATG')
        i = 100000
        continue

    i = atg + 1

    minStop = min(stops)
    if minStop-atg < SHORT :
        #print('Too short')
        continue

    # AGGAGG
    aggagg = findPattern('AGGAGG', seq, atg-13, atg-9,1,5)

    if aggagg == -1 :
        continue

    zz+=1
    print('\nORF #', zz, end =' ')
    print('Shine: ', aggagg, end=' ')
    print('ATG', atg, end = ' ')
    #print('---------')
    print('Stop at:', minStop)
```

ORF # 1 Shine: 441 ATG 451 Stop at: 592

ORF # 2 Shine: 3734 ATG 3747 Stop at: 4776

ORF # 3 Shine: 5808 ATG 5820 Stop at: 6363

ORF # 4 Shine: 7950 ATG 7962 Stop at: 8697

ORF # 5 Shine: 11313 ATG 11324 Stop at: 11588

ORF # 6 Shine: 12921 ATG 12934 Stop at: 13177

ORF # 7 Shine: 13269 ATG 13282 Stop at: 14149

ORF # 8 Shine: 14120 ATG 14132 Stop at: 14864

ORF # 9 Shine: 14278 ATG 14289 Stop at: 14457

ORF # 10 Shine: 14887 ATG 14899 Stop at: 15805