A Primer to String Algorithms

SBS BIOHACKATHON || PYTHON

Scope

1. Recap From Last Week
2. Challenge 5: Inexact String Search
3. Reading And Writing From/To Files
4. Challenge 6: Edit Distance
5. Where to go from here?

Recap; For Loops

```
seq = ‘AGACAG’

for i in range(len(seq)):
    prints: 0,1,2,3,4,5
    To print base:
    print (seq[i]) -> prints: A,G,A,C,A,G

for base in seq:
    prints: A,G,A,C,A,G
```

Challenge 3
Hamming Distance

In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different.

Example:

```
AGGTGTTCGCTG
|| || ||| ||
AGCTGATCGATG
```

Hamming Distance Score = 3

If seqA[i] != seqB[i]

Challenge 4
Exact String Search

Your Task was to write a Function to query a motif pattern (len=5) in a DNA sequence and return its index if found.

```
Text    = ‘AGTCGATCGATGCGTCGATGCTAGCTGATCGAT’

Pattern = ‘TCGAT’
```

```
AGTCTATCGATGCGTCGATGCTAGCTGATCGAT
  |||||
  TCTAT
```

check: if text[**2**:7] == pattern

return True

return index = 2

Challenge 5
Inexact String Search

Your Task is to write a Function to query a motif pattern (len=5) in a DNA sequence, **tolerating a mismatch of 1bp per 5 nucleotide length,** and return its index if found.

Example:

```
Text    = ‘AGTCGATCGATGCGTCGATGCTAGCTGATCGAT’
Pattern = ‘TTGAT’
```

```
AGTCTATCGATGCGTCGATGCTAGCTGATCGAT
  | |||
  TTGAT
```

check: if text[**2**:7] == pattern??
Is this correct tho? How do we check for One mismatch?

return True

return index = 2

GAT

```
AGCTCTAT
 GAT
     GAT
```

Reading And Writing From/To Files

```
# To Write
with open(csv_file, 'w') as f:
    ## Insert Code Here
    f.write(f"Message: {}\n")

# To Write
open_file = open(csv_file, 'w')
open_file.write(f"Message: {}\n")
open_file.close()
```

Reading And Writing From/To Files

```
# To Read
with open(csv_file, 'r') as f:
    ## Insert Code Here
    for line in f:
        line = line.strip('\n')
        col = line.split(',')
        ## Do Something

# To Read
open_file = open(csv_file, 'r')
for line in open_file:
    line = line.strip('\n')
    col = line.split(',’)
    ## Do Something
```

Nested Loops

```
seq = ‘AGACAG’
for base in seq:
        prints: A,G,A,C,A,G

list_of_sequences = [‘AGACAG’, ‘GAGTC’, ‘AGTGAC’]
for seq in list_of_sequences:
        for base in seq:
                prints: A,G,A,C,A,G
                prints: G,A,G,T,C
                prints: A,G,T,G,A,C
```

Nested Loops

```
numbers = [1,2,4,5]
for nu in numbers:
       prints: 1,2,4,5

list_of_numbers = [[1,2,4,5], [2,3,5,5], [1,1,4,4]]
for numbers in list_of_numbers :
       for nu in numbers:
              prints: 1,2,4,5
              prints: 2,3,5,5
              prints: 1,1,4,4

list_of_numbers = [[1,2,4,5],
                   [2,3,5,5],
                   [1,1,4,4]]
```

Edit Distance

A way of quantifying how dissimilar two strings (e.g., DNA Sequences) are to one another, that is measured by counting the minimum number of operations required to transform one string into the other

Levenshtein distance operations are the deletion, insertion, or substitution of a character in the string.

```
Query: AGCTCG
Hit  : AGCTAG
```

		A	G	C	T	C	G
	0	1	2	3	4	5	6
A	1	0	1	2	3	4	5
G	2	1	0	1	2	3	4
C	3	2	1	0	1	2	3
T	4	3	2	1	0	1	2
A	5	4	3	2	1	1	2
G	6	5	4	3	2	2	1

Dynamic Programming

Dynamic Programming is a technique in computer programming that helps to efficiently solve a class of problems that have overlapping subproblems and optimal substructure property.

In this case, we build a matrix (2 by 2) to solve for edit distance.

		A	G	C	T	C	G
	0	1	2	3	4	5	6
A	1						
G	2						
C	3						
T	4						
A	5						
G	6						

Edit Distance

4 situations:

1. Match (Move diagonally)
2. Substitution (Move diagonally + 1)
3. Insertion (Move right + 1)
4. Deletion (Move down + 1)

Find the lowest score

```
Query: AGCTCG
Hit  : AGCTAG
```

		A	G	C	T	C	G
	0	1	2	3	4	5	6
A	1	0	1	2	3		
G	2	1					
C	3						
T	4						
A	5						
G	6						

		T
	3	4
A	2	

Sub = 4
Insetion = 3
Deletion = 5

AGCT
A

Edit Distance

4 situations:

1. Match (Move diagonally)
2. Substitution (Move diagonally + 1)
3. Insertion (Move right + 1)
4. Deletion (Move down + 1)

Find the lowest score

Query: AGCTCG
Hit : AGCTAG

		A	G	C	T	C	G
	0	1	2	3	4	5	6
A	1	0	1	2			
G	2	1	0				
C	3						
T	4						
A	5						
G	6						

Edit Distance

4 situations:

1. Match (Move diagonally)
2. Substitution (Move diagonally + 1)
3. Insertion (Move right + 1)
4. Deletion (Move down + 1)

Find the lowest score

```
Query: AGCTCG
Hit  : AGCTAG
```

		A	G	C	T	C	G
	0	1	2	3	4	5	6
A	1	0	1	2			
G	2	1	0				
C	3						
T	4						
A	5						
G	6						

		A	G	C	T	C	G
	0	1	2	3	4	5	6
A	1	0	1	2	3	4	5
G	2	1	0	1	2	3	4
C	3	2	1	0	1	2	3
T	4	3	2	1	0	1	2
A	5	4	3	2	1	1	2
G	6	5	4	3	2	2	1

Edit Distance

4 situations:

1. Match (Move diagonally)
2. Substitution (Move diagonally + 1)
3. Insertion (Move right + 1)
4. Deletion (Move down + 1)

Find the lowest score

```
Query: TGATA
Hit  : TGGACT
```

		T	G	A	T	A
	0	1	2	3	4	5
T	1	0				
G	2					
G	3					
A	4					
C	5					
T	6					

		T	G	A	T	A
	0	1	2	3	4	5
T	1	0	1	2	3	4
G	2	1	0	1	2	3
G	3	2	1	1	2	3
A	4	3	2	1	2	2
C	5	4	3	2	2	3
T	6	4	3	3	2	3

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

A*STAR Genome Institute of Singapore

NalaGenetics

Agency for Science, Technology and Research SINGAPORE

Background

NTU SBS Graduate 2020

Bioinformatics Specialist

- A*STAR Genome Institute of Singapore (GIS)
- Nalagenetics

Part Time Masters in BioMedical Data Science

Where to Next?

Rosalind Bioinformatics Challenge

Coursera Bioinformatics (UC San Diego / John Hopkins)

William Fiset Data Structures (8 Hours)

RNA-seq Youtube Code Along

Find a lab attachment

Kaggle (For machine learning; data analyst / scientist)

Build a github repository!

Code code code!