22BIO201 Intelligence of Biological Systems 1

Lab Sheet 4

Contents

Problem 1 : Compute Distance Between Pattern And Strings

Find the distance between a pattern and a set of strings.

Given: A DNA string Pattern and a collection of DNA strings *Dna*.

Return: *DistanceBetweenPatternAndStrings(Pattern, Dna).*

```
DistanceBetweenPatternAndStrings(Pattern, Dna)
    k ← |Pattern|
    distance ← 0
    for each string Text in Dna
        HammingDistance ← ∞
        for each k-mer Pattern' in Text
            if HammingDistance > HammingDistance(Pattern, Pattern')
                HammingDistance ← HammingDistance(Pattern, Pattern')
        distance ← distance + HammingDistance
    return distance
```

Sample Dataset

AAA

TTACCTTAAC GATATCTGTC ACGGCGTTCG CCCTAAAGAG CGTCAGAGGT

Sample Output

5

Problem 2 : Brute Force Motif Search - Implanted Motif Problem

Implement Brute Force Motif Search for a set of DNA strings.

Given a collection of strings Dna and an integer d, a k-mer is a **(k,d)-motif** if it appears in every string from Dna with at most d mismatches. The following algorithm finds (k,d)-motifs.

```
MOTIFENUMERATION(Dna, k, d)
    Patterns ← an empty set
    for each k-mer Pattern in Dna
      for each k-mer Pattern' differing from Pattern by at most d
       mismatches
        if Pattern' appears in each string from Dna with at most d
        mismatches
          add Pattern' to Patterns
    remove duplicates from Patterns
    return Patterns
```

Implanted Motif Problem

Implement MotifEnumeration (shown above) to find all *(k, d)-motifs* in a collection of strings.

Given: Integers *k and d*, followed by a collection of strings *Dna*.

Return: All *(k, d)-motifs* in Dna.

Sample Dataset

```
3 1
ATTTGGC
TGCCTTA
```

CGGTATC
GAAAATT

Sample Output

ATA ATT GTT TTT

Problem 3: Scoring Motifs

Given a set of *'t'* DNA Strings, display a Motif Matrix and calculate the corresponding Count matrix and Profile matrix. Use the profile matrix to form the Consensus string.

Dataset : Use NF-xB binding sites and form consensus "**TCGGGGATTTCC"**

Problem 4: Find a Profile-most Probable k-mer in a String

Given a profile matrix *Profile*, we can evaluate the probability of every *k*-mer in a string *Text* and find a ***Profile*-most probable** *k*-mer in *Text*, i.e., a *k*-mer that was most likely to have been generated by *Profile* among all *k*-mers in *Text*.

For example, **ACGGGGATTACC** is the *Profile*-most probable 12-mer in GGT**ACGGGGATTACC**T. Indeed, every other 12-mer in this string has probability 0.

In general, if there are multiple *Profile*-most probable *k*-mers in *Text*, then we select the first such *k*-mer occurring in *Text*.

Profile-most Probable *k*-mer Problem

Find a Profile-most probable k-mer in a string.

Given: A string *Text*, an integer *k*, and a 4 × *k* matrix *Profile*.

Return: A *Profile*-most probable *k*-mer in *Text*. (If multiple answers exist, you may return any one.)

Sample Dataset

```
ACCTGTTTATTGCCTAAGTTCCGAACAAACCCAATATAGCCCGAGGGCCT
5
0.2 0.2 0.3 0.2 0.3
0.4 0.3 0.1 0.5 0.1
0.3 0.3 0.5 0.2 0.4
0.1 0.2 0.1 0.1 0.2
```

Sample Output

```
CCGAG
```