CS 3824
Solutions to Homework Assignment 1

Per Emil Johan Bengtsson

September 7, 2018

[25] 1. Jones and Pevzner problem 4.12.

Give pseudocode for your algorithm, along with an English explanation of how it works. Determine a O bound on its worst-case time complexity as a function of the lengths of T and s. (Use $|T|$ and $|s|$ for these lengths.)

$T = t_1, t_2, ..., t_n$

FIRST_OCCURRENCE(T, s)
l = length of s
for each letter t_i in T, $i \geq l$:
    $substr = t_{i-l+1}, t_{i-l+2}, ..., t_i$
    if $substr$ is equal to s:
        return $i - l + 1$
output "no occurrence"
return $Null$

The algorithm starts looking at the first characters in T (as many as there are in s), and compares them with s. If they are not the same as the characters in s, the loop starts over and looks at $|s|$ new characters in T, starting with the second character instead of the first this time. If a match is found, the index of the first letter in the match is returned. If no match is found in T, $Null$ is returned.

In the worst case, which is when s does not occur in T, the algorithm compares $|T| - |s| + 1$ different strings of length $|s|$ with s. So the worst case time complexity of the algorithm is $O(|T| * |s|)$

---

[25] 2. Jones and Pevzner problem 4.13.

Give pseudocode for your algorithm, along with an English explanation of how it works. Determine a O bound on its worst-case time complexity as a function of the lengths of T and s and of k.

$T = t_1, t_2, ..., t_n$

FIRST_OCCURRENCE(T, s, k)
l = length of s
for each letter t_i in T, $i \geq l$:
    $substr = t_{i-l+1}, t_{i-l+2}, ..., t_i$
    if HAMMING_DISTANCE($substr$, s) $\leq k$:
        return $i - l + 1$
        output $substr$
output "no occurrence"
return $Null$

HAMMING_DISTANCE(s', s)
sum = 0
for each character at position i in s and s':
    if $s'_i \neq s_i$:
        sum = sum + 1
return sum

The algorithm starts looking at the first characters in T (as many as there are in s), and calculates their Hamming distance to s. If the Hamming distance is more than or equal to k characters in s, the loop starts over and looks at $|s|$ new characters in T, starting with the second character instead of the first this time. If the calculated Hamming distance between the two strings is less than k, the index of the first letter in the substring of T is returned and the substring is also printed. If no match is found in T, $Null$ is returned.

In the worst case, which is when s does not have a Hamming distance of less than k for any substring of T, the algorithm calculates the Hamming distance for $|T| - |s| + 1$ different strings of length $|s|$. The HAMMING_DISTANCE-function has a time complexity of $O(|s|)$; so the worst case time complexity of the whole algorithm is $O(|T| * |s|)$

---

[30] 3. A problem inspired by Jones and Pevzner problem 4.14.

If you have a DNA, RNA, or protein sequence (string) X and k is a positive integer, then a *k-mer of X* is any substring of X of length exactly k.

FASTA is a simple file format for biological sequences. You can find an example FASTA file

`Escherichia_coli_genome.fasta`

under Resources on the course Web site. That file contains a complete genome for a substrain of the bacterium *Escherichia coli* (E. coli). If you investigate the file format, you

```
Sequence length is 4641652.
k value is 1.
A 1142742
C 1180091
G 1177437
T 1141382
Maximum count is 1180091:
C 1180091
Minimum count is 1141382:
T 1141382
```

Figure 1: Expected output when $k = 1$.

will see how to obtain the complete genome sequence by concatenating the second through the last lines of the file.

You are to write a program in a language of your choice to analyze the k-mer content of the E. coli genome. (The posted solution will be a Python 3 program; Python 3 is an easy language to use for this assignment.) Your program must take the value of k as a parameter (k may default to 1, if you like). It must extract all the k-mers from the genome and count how many of each k-mer occurs. Also, compute the maximum and minimum counts greater than 1 and which k-mers meet those counts. As a result, your program will be able to generate the expected reports in Figures 1 and 2.

Once you have your program that can generate the two sample reports, generate the corresponding report for $k = 8$. In the `.tar` file you submit, include your report for $k = 8$ and your program source file(s). In your PDF solution file, specify the language you used and briefly explain your strategy for your program.

I used Python 3 to write my program. The program takes the k value as an input argument, if none is provided k defaults to 1. It then looks at all the substrings of length k in the Escherichia coli genome and stores a counter for each of them in a dictionary. For each substring that is looked at, the counter for that specific substring gets incremented. Then it loops through all of the substrings to print the number of times they occur in the genome and it also figures out which substrings have the maximum and minimum counts and prints them.

```
Sequence length is 4641652.
k value is 2.
AA 338006
AC 256773
AG 238013
AT 309950
CA 325327
CC 271821
CG 346793
CT 236149
GA 267384
GC 384102
GG 270252
GT 255699
TA 212024
TC 267395
TG 322379
TT 339584
Maximum count is 384102:
GC 384102
Minimum count is 212024:
TA 212024
```

Figure 2: Expected output when $k = 2$.