

Брянцев Всеволод Александрович

ПММ, 4 курс, 61 группа

Отчет по лабораторной работе №6

Вариант 4.

### Задание 1.

В базе данных GenBank найти нуклеотидные последовательности в формате Fasta в соответствии с вариантом.

4. Human T-lymphotropic virus 3 (Т-лимфотропный вирус человека 3 типа), Simian T-lymphotropic virus 3 (Т-лимфотропной вирус обезьян)

>KY493191.1 Human T-lymphotropic virus 3 isolate NKO39 gag protein gene, partial cds

```
CGATCACCTCTCTGTATCTGCCCTTCCCTAATCAGAACCATGGGAAAGACTTATAGCTCTCCGGTAAACC
STATCCCCAAAGCCCCAAAAGGGCTAGCAATTCACCACTGGCTAAACTTTCTCCAAGCCGCGTACCGACT
GCAGCCAGGGCCCTCTGAATTTGACTTCCATCAGTTACGGAAGTTCTAAAACTAGCCATCAAGACCCCG
GTGTGGTTAAATCCCATTAACTCTGTCCTTGCCGAGCTCATTCCTCAAGAACTACCCGGGCAGGGTCC
ATGAAATAGTAGCCATCCTAATTCAAGAGACCCCTGCGCGCGAGGCGCCCCATCGGCCCCACCAGCAGA
TGACCCTCAAAAGCCTCCACCCTACCCCGAGCATGCACAGGTAGAACCTCAGTGCCTCCCTGTCCTTCAC
CCTCACGGGGCCCCAGCCACTCACCGCCCCT
```

>EU152290.1 Simian T-lymphotropic virus 3 strain Cag9748NL tax protein (tax) gene, partial cds

```
CTACGGGTACCCTGTCTACGTTTTTCGGCGACTGTGTACAGGCCGATTGGTGCCCCATTTCTGGGGGGCTT
TGTTCCGCTCGGCTACACCGCCATGCCTTGCTGGCCACGTGCCCCGAACATCAGATTACCTGGGACCCCA
TCGATGGACGCGTTGTCAGCTCAGCTCTACAATA
```

### Задание 2.

Написать программу поиска наиболее часто встречающегося k-мера в строке, k каждый раз вводится любое, меньшее длины строки. Тестовую последовательность для примера брать из Задания 1. Вывести наиболее часто встречающуюся строку длины 7.

Код программы:

```
import 'dart:io';

Future<String> readFile(String filePath) async {
  return await File(filePath).readAsString();
}
```

```

}

List<MapEntry<String, int>> frequentWords(String str, int k) {
    Map<String, int> kmerCounts = {};
    int maxCount = 0;

    for (int i = 0; i <= str.length - k; i++) {
        String kmer = str.substring(i, i + k);
        kmerCounts[kmer] = (kmerCounts[kmer] ?? 0) + 1;
        maxCount = maxCount > kmerCounts[kmer]! ? maxCount : kmerCounts[kmer]!;
    }

    return kmerCounts.entries.where((entry) => entry.value == maxCount).toList();
}

Future<void> main() async {
    String text1 = await readFile('bin\\input1.txt');
    String text2 = await readFile('bin\\input2.txt');

    stdout.write("Введите значение k: ");
    int k = int.parse(stdin.readLineSync()!);

    List<MapEntry<String, int>> result1 = frequentWords(text1, k);
    List<MapEntry<String, int>> result2 = frequentWords(text2, k);

    print("\nНаиболее частые $k-меры в первой последовательности:");
    if (result1.isNotEmpty) {
        print("Количество вхождений: ${result1[0].value}");
        for (var entry in result1) {
            print(entry.key);
        }
    }

    print("\nНаиболее частые $k-меры во второй последовательности:");
    if (result2.isNotEmpty) {
        print("Количество вхождений: ${result2[0].value}");
        for (var entry in result2) {
            print(entry.key);
        }
    }
}

```

Две найденные в Задании 1 последовательности содержатся в файлах input1 и input2:

input1:

```

CGATCACCTCTCTGTATCTGCCCTTCCCTAATCAGAACCATGGGAAAGACTTATAGCTCTCCGGTAAACCCTA
TCCCCAAAGCCCCAAAAGGGCTAGCAATTCACCACTGGCTAAACTTTCTCCAAGCCGCGTACCGACTGCA
GCCAGGGCCCTCTGAATTTGACTTCCATCAGTTACGGAAGTTCCTAAAACTAGCCATCAAGACCCCGGTGT

```

GGTTAAATCCCATTAATTACTCTGTCCTTGCCGAGCTCATTCCTCAAGAACTACCCGGGCAGGGTCCATGAAA  
TAGTAGCCATCCTAATTCAAGAGACCCCTGCGCGAGGCGCCCCATCGGCCCCACCAGCAGATGACCCT  
CAAAAGCCTCCACCCTACCCGAGCATGCACAGGTAGAACCTCAGTGCCTCCCTGTCCTTCACCCTCACGG  
GGCCCCAGCCACTACCGCCCCCT

input2:

CTACGGGTACCCTGTCTACGTTTTCGGCGACTGTGTACAGGCCGATTGGTGCCCCATTTCTGGGGGGCTTT  
GTTCCGCTCGGCTACACCGCCATGCCTTGCTGGCCACGTGCCCCGAACATCAGATTACCTGGGACCCCATC  
GATGGACGCGTTGTCAGCTCAGCTCTACAATA

Результат:

Для k = 4:

```
Введите значение k: 4

Наиболее частые 4-меры в первой последовательности:
Количество вхождений: 10
CCCT
CCCC

Наиболее частые 4-меры во второй последовательности:
Количество вхождений: 4
CTAC
```

Для k = 7:

```
Введите значение k: 7

Наиболее частые 7-меры в первой последовательности:
Количество вхождений: 2
CCCCAA
TAGCCAT
AGCCATC
AGACCCC
CTGTCCT
TGTCTT
GGCCCCA
ACCTCA

Наиболее частые 7-меры во второй последовательности:
Количество вхождений: 2
GTGCCCC
TCAGCTC
```