Лабораторная работа № 6

Поиск наиболее часто встречающейся подстроки в строке

1 Алгоритм поиска наиболее часто встречающейся подстроки в строке

Пусть задано конечное множество «A», которое будет называться «алфавитом», а его элементы «буквами». Обычно сразу же требуется, чтобы на «A» было задано упорядочение, что эквивалентно требованию нумерации букв. На самом деле, нумерация в реально используемых алфавитах может легко изменяться в зависимости от поставленной задачи.

В задачах биоинформатики чаще всего имеют дело любо с четырехбуквенным алфавитом (ДНК, РНК) либо с двадцатибуквенным алфавитом (белок). В этой лабораторной работе мы будем рассматривать 4-буквенный алфавит, состоящий из букв A, T, G, C. Набор букв из такого алфавита будем называть последовательностью. Строки — это произвольные последовательности, составленные из букв, причем любая буква может встречаться в последовательности любое число раз. Например, строка

s= ACGTTGCATGTCGCATGATGCATGAGAGCT

Рассматриваемые нами подпоследовательности должны быть содержать символы в том же порядке, что и исходная последовательность, например если v=ATTGCTA, то AGCA и ATTA являются ее подпоследовательностями, а TGTT и TCG — не являются.

Подстрока — это подпоследовательность, составленная из букв строки, идущих подряд.

Подстроку длины k будем называть *k-мером*.

Рассмотрим задачу о нахождении наиболее часто повторяющиеся подстроки в геноме.

1. Первая часть этой задачи состоит в том, чтобы в строке длины L найти подстроку длины k. Мы будем использовать термин k-мер.

Задача 1. Посчитать количество k-меров в строке.

Входные данные: строка и k-мер.

Выходные данные: число этих k-меров в строке.

Алгоритм:

PatternCount(Text, Pattern)
count = 0
for i = 0 to |Text|- |Pattern|
if Text(i, |Pattern|) = Pattern
count = count + 1
return count

Пример работы программы подсчета количества k-меров в строке.

Входные данные: строка = "ACAACTATGCATACTATCGGGAACTATCCT" и k-мер="ACTAT" .

Выходные данные: число этих k-меров в строке:

PatternCount("ACTAT" , "ACAACTATGCATACTATCGGGAACTATCCT") = 3

Задача 2. Нахождение наиболее часто встречающихся k-меров в строке.

Например, рассмотрим строку

CG**ATATA**TCC**ATA**G

Наиболее часто встречающийся 3-мер в ней — это ATA.

Алгоритм для нахождения наиболее часто встречающихся k-меров в строке вычисляет, сколько раз появляется каждый k-мер в строке, а затем выбирает k-мер, который происходит больше всего. Чтобы реализовать этот алгоритм сгенерируем массив следующего вида:

i	0	1	2	3	4	5	6	7	8	9	10	11		
Count[i]	1	1	3	2	3	2	1	1	1	1	3	1		
Text	C	G	A	T	A	T	A	T	C	C	A	T	A	G

Используя Python мы можем создать словарь (dict) в котором будет храниться рассмотренный массив, где Count[i] количество появлений i-го k-мера в строке Text.

Для вычисления Count мы можем использовать уже рассмотренную функцию PatternCount(Text, Pattern).

Итак, мы имеем функцию CountDict(Text, k).

Алгоритм:

CountDict(Text, k)

PatCount = \{\}
for i = 0 to |Text|- |Pattern|
Pattern=Text[i:i+k]
PatCount = PatternCount(Pattern, Text)
return PatCount

Входные данные: строка и длина подстроки k.

Выходные данные: словарь, где каждому k-меру сопоставлено количество его вхождений в строку.

Пример работы программы CountDict(Text, k)

Входные данные: строка = "CGATATATCCATAG" и k=3.

Выходные данные: массив:

{0: 1, 1: 1, 2: 3, 3: 2, 4: 3, 5: 2, 6: 1, 7: 1, 8: 1, 9: 1, 10: 3, 11: 1}

Чтобы определить наиболее часто встречающиеся k-меры в строке, мы просто должны найти максимальное значение из всех значений словаря. Python имеет встроенную функцию называемую values(), который возвращает список, содержащий значения словаря. Поэтому мы можем вычислить максимум всех значений в данном списке, используя следующую функцию поиска максимального значения values(). Функцию, которая выводит все наиболее часто встречающиеся k-меры в троке назовем FrequentWords(Text, k).

Чтобы определить наиболее часто встречающиеся k-меры в строке, мы просто должны найти максимальное значение из всех значений словаря. Python имеет встроенную функцию называемую values(), который возвращает список, содержащий значения словаря. Поэтому мы можем вычислить максимум всех значений в данном списке, используя следующую функцию поиска максимального значения values(). Функцию, которая выводит все наиболее часто встречающиеся k-меры в строке назовем FrequentWords(Text, k).

Пример работы программы FrequentWords(Text, k)

Входные данные: строка = "ACGTTGCATGTCGCATGATGCATGAGAGCT" и k=4 .

Выходные данные:

GCAT CATG GCAT CATG GCAT CATG

Мы замечаем, что 4-меры из предыдущего примеры повторяются. Следовательно, дальнейший шаг - это удаление повторяющихся k-меров.

Запишем алгоритм поиска наиболее часто встречающегося k-мера в строке с удалениемвсех повторяющихся k-меров.

Алгоритм поиска наиболее часто встречающегося k-мера в строке

FrequentWords(Text, k)
FrequentPatterns = an empty set
for i = 0 to |Text|- k
Pattern = the k-mer Text(i, k)
Count(i) = PatternCount(Text, Pattern)
maxCount = maximum value in array Count
for i = 0 to |Text|- k
if Count(i) = maxCount
add Text(i, k) to FrequentPatterns
remove duplicates from FrequentPatterns
return FrequentPatterns

2 Задания

Задание 1.

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

Варианты:

1. Homo sapiens hemoglobin (гемоглобин человека), Mus musculus hemoglobin (гемоглобин домовой мыши)

2. Populus tremula papain (Папаин осины), Culex quinquefasciatus procathepsin L3 (прокатепсин L3 южного домашнего комара)

3. Myxine glutinosa calmodulin (Кальмодулин рыбы-ведьмы), Cricetulus griseus calmodulin (Кальмодулин китайского хомяка)

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

5. Garlic virus A (вирус чеснока А), Garlic virus B (вирус чеснока В)

6. Bacteriophage SV14 single-stranded binding protein (вирус Enterobacteria phage SV14)

7. Ictalurid herpesvirus 2 (вирус Ictalurid herpesvirus 2), Salmonid herpesvirus 2 (вирус Salmonid herpesvirus 2)

8. Datura yellow vein virus, Taro vein chlorosis virus

9. Carajas virus, Maraba virus

10. Porcine reproductive and respiratory syndrome virus, Duck hepatitis A virus

11. Zea mays hemoglobin, Paramecium caudatum hemoglobin

12. Octodon degus insulin (инсулин дегу), Oryctolagus cuniculus New Zealand White insulin (инсулин дикого новозеландского кролика)

13. Chicken nerve growth factor (куриный фактор роста нервов), Anairetes alpinus bolivianus nerve growth factor (NGF) gene, exon 4 (4 экзон фактора роста нервов птиц, семейства тиранновые мухоловки)

14. Oncorhynchus mykiss thrombin mRNA (тромбин радужной форели, мРНК),

15. Pinellia ternata agglutinin gene (ген агглютинина растения, семейства ароидные)

16. Drosophila melanogaster cytochrome c proximal

17. Halocynthia roretzi mRNA for claudin

18. Synthetic mouse epidermal growth factor gene

19. Pig mRNA for epidermal growth factor

20. Human liver/bone/kidney-type alkaline phosphatase (ALPL) gene, exon 3

21. Danio rerio thrombopoietin (thpo), mRNA

22. Human cystic fibrosis transmembrane conductance regulator (CFTR) gene, exon 5

23. Human renin gene, exon 1

24. Canis familiaris mRNA for frataxin

25. Sowthistle yellow vein virus

26. Ovis aries myostatin (MSTN) gene, exon 3

27. Gallus gallus isolate 4767 myostatin (MSTN) gene, MSTN-Q allele, exon 3

28. Lophonetta specularioides alticola voucher UAM:REW 721 lamin A (LMNA) gene, exons 3

29. Bacteriophage nt-1 tail tube protein (вирус Vibrio phage nt-1)

30. Cynops pyrrhogaster thrombin mRNA (тромбин огненнобрюхого тритона, мРНК)

31. Zephyranthes candida agglutinin gene (ген агглютинина белой водяной лилии)

32. N.tabacum mRNA for cytochrome b5

33. Canis lupus familiaris claudin 2 mRNA

34. Synthetic human epidermal growth factor gene

35. Artificial gene for human epidermal growth factor

36. R.norvegicus gene encoding alkaline phosphatase, exon 3 and joined CDS

37. Gallus gallus thrombopoietin (Tpo) mRNA

38. Human cystic fibrosis transmembrane conductance regulator (CFTR) gene, exon 19

39. Mouse Ren1 gene for renin exon 1

40. Arabidopsis thaliana frataxin mRNA

41. Cowpea mosaic virus

42. Gallus gallus isolate 4767 myostatin (MSTN) gene, MSTN-Q allele, exon 3

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.