

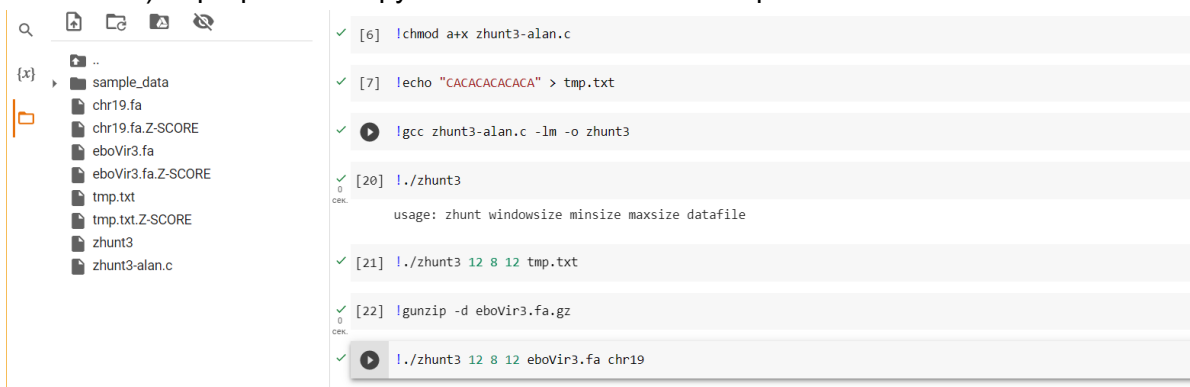
ОТЧЕТ

О выполнении Домашнего Задания 4

Выполнила Чередова Диана

Ход Работы

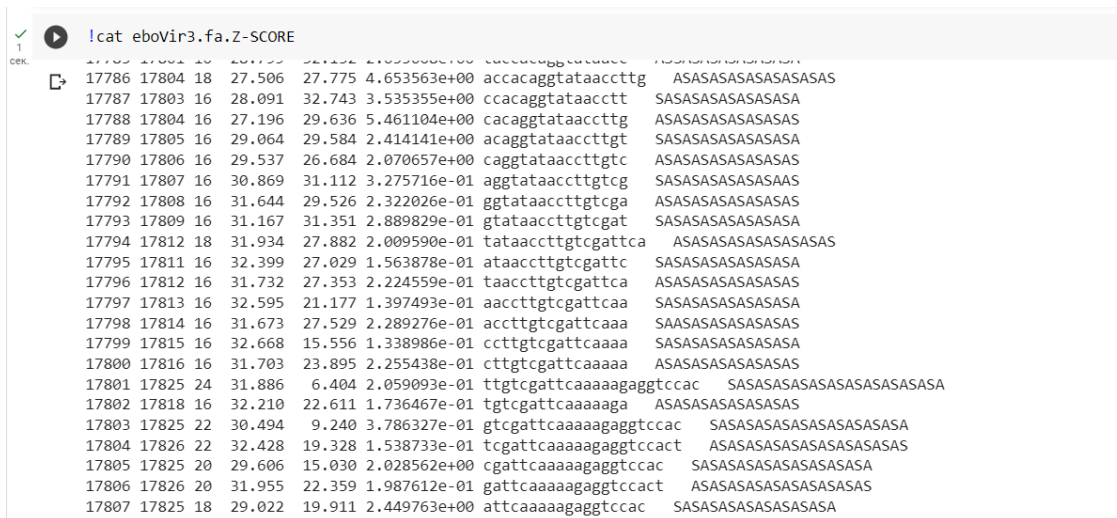
- 1) С прекрасного сайта <https://hgdownload.soe.ucsc.edu/downloads.html> Мною был скачан геном вируса Эбола. На семинаре была дана рекомендация о том, что лучше использовать данные для вируса, так как они быстрее обрабатываются.
- 2) Программа загружена в коллаб и готова к работе!



The screenshot shows a terminal window with a file explorer on the left. The file explorer displays a directory structure with files like 'sample_data', 'chr19.fa', 'chr19.fa.Z-SCORE', 'eboVir3.fa', 'eboVir3.fa.Z-SCORE', 'tmp.txt', 'tmp.txt.Z-SCORE', 'zhunt3', and 'zhunt3-alan.c'. The terminal window shows the following commands and their outputs:

```
[6] !chmod a+x zhunt3-alan.c
[7] !echo "CACACACACACA" > tmp.txt
[20] !./zhunt3
usage: zhunt windowsize minsize maxsize datafile
[21] !./zhunt3 12 8 12 tmp.txt
[22] !gunzip -d eboVir3.fa.gz
!./zhunt3 12 8 12 eboVir3.fa chr19
```

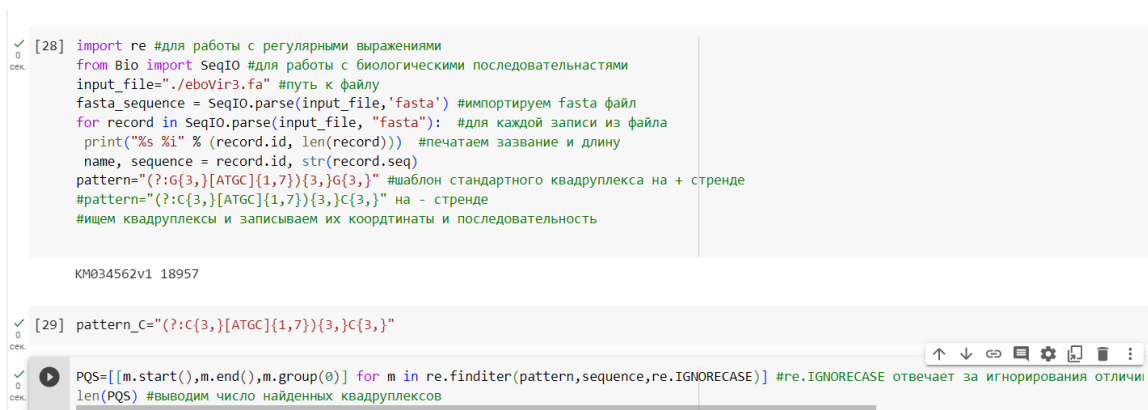
- 3) Данные обработаны и представлены в виде таблицы с полученными участками Z-ДНК.



The screenshot shows a terminal window with the command `!cat eboVir3.fa.Z-SCORE` and its output. The output is a table with columns representing genomic coordinates and Z-scores. The first few rows are:

chr	start	end	score	sequence
17786	17804	18	27.506	27.775 4.653563e+00 accacaggtataaccttg ASASASASASASASAS
17787	17803	16	28.091	32.743 3.535355e+00 ccacaggtataacctt SASASASASASASASA
17788	17804	16	27.196	29.636 5.461104e+00 cacaggtataaccttg ASASASASASASASAS
17789	17805	16	29.064	29.584 2.414141e+00 acaggtataaccttgt SASASASASASASASA
17790	17806	16	29.537	26.684 2.070657e+00 caggtataaccttgtc ASASASASASASASAS
17791	17807	16	30.869	31.112 3.275716e-01 aggtataaccttgtcg SASASASASASASAS
17792	17808	16	31.644	29.526 2.322026e-01 ggataaccttgtcga ASASASASASASASAS
17793	17809	16	31.167	31.351 2.889829e-01 gtataaccttgtcgat SASASASASASASASA
17794	17812	18	31.934	27.882 2.009590e-01 tataaccttgtcgattca ASASASASASASASAS
17795	17811	16	32.399	27.029 1.563878e-01 ataaccttgtcgattc SASASASASASASASA
17796	17812	16	31.732	27.353 2.224559e-01 taaccttgtcgattca ASASASASASASASAS
17797	17813	16	32.595	21.177 1.397493e-01 aaccttgtcgattcaa SASASASASASASASA
17798	17814	16	31.673	27.529 2.289276e-01 accttgtcgattcaaa SASASASASASASASA
17799	17815	16	32.668	15.556 1.338986e-01 cttgtcgattcaaaa SASASASASASASASA
17800	17816	16	31.703	23.895 2.255438e-01 cttgtcgattcaaaaa ASASASASASASASAS
17801	17825	24	31.886	6.404 2.059093e-01 ttgtcgattcaaaaagggtccac SASASASASASASASAS
17802	17818	16	32.210	22.611 1.736467e-01 tgcgattcaaaaaga ASASASASASASASAS
17803	17825	22	30.494	9.240 3.786327e-01 gtcgattcaaaaagggtccac SASASASASASASASAS
17804	17826	22	32.428	19.328 1.538733e-01 tcgattcaaaaagggtccact ASASASASASASASAS
17805	17825	20	29.606	15.030 2.028562e+00 cgattcaaaaagggtccac SASASASASASASASAS
17806	17826	20	31.955	22.359 1.987612e-01 gattcaaaaagggtccact ASASASASASASASAS
17807	17825	18	29.022	19.911 2.449763e+00 attcaaaaagggtccac SASASASASASASASAS

- 4) Ищем квадруплексы по шаблону



The screenshot shows a terminal window with the following Python code:

```
[28] import re #для работы с регулярными выражениями
from Bio import SeqIO #для работы с биологическими последовательностями
input_file="./eboVir3.fa" #путь к файлу
fasta_sequence = SeqIO.parse(input_file,'fasta') #импортируем fasta файл
for record in SeqIO.parse(input_file, "fasta"): #для каждой записи из файла
    print("%s %i" % (record.id, len(record))) #печатаем зазвание и длину
    name, sequence = record.id, str(record.seq)
    pattern="(?<G{3,}[ATGC]{1,7}){3,}G{3,}" #шаблон стандартного квадруплекса на + стренде
    #pattern="(?<C{3,}[ATGC]{1,7}){3,}C{3,}" на - стренде
    #ищем квадруплексы и записываем их координаты и последовательность

KM034562v1 18957

[29] pattern_c="(?<C{3,}[ATGC]{1,7}){3,}C{3,}"

PQS=[m.start(),m.end(),m.group(0)] for m in re.finditer(pattern,sequence,re.IGNORECASE) #re.IGNORECASE отвечает за игнорирования отличии
len(PQS) #выводим число найденных квадруплексов
```

Ответы на вопросы:

- 1) Так как файл маленький сам по себе, то при установке порогового значения 300 участков Z-ДНК, больших порогового, осталось 2:

```
import pandas as pd
data=pd.read_csv("eboVir3.fa.Z-SCORE", skiprows=1, names=["Start","End","1","2","3","Score","Seq","4"], delim_whitespace=True)
data.loc[data['Score'] >= 300]
```

	Start	End	1	2	3	Score	Seq	4
14680	14681	14697	16	22.181	32.253	332.9764	tattttcacgcacgcc	ASASASASASASASAS
14682	14683	14699	16	22.119	32.252	357.1333	ttttcacgcacgcoga	ASASASASASASASAS

Поэтому пороговое значение выбрала 150.
Для такого случая участков Z-ДНК стало 24.

```
import pandas as pd
data=pd.read_csv("eboVir3.fa.Z-SCORE", skiprows=1, names=["Start","End","1","2","3","Score","Seq","4"], delim_whitespace=True)
data.loc[data['Score'] >= 150]
```

	Start	End	1	2	3	Score	Seq	4
6439	6440	6456	16	22.806	46.898	169.2888	gtgccggtatgtgcac	SASASASASASASASA
8634	8635	8657	22	22.908	46.198	152.3017	aggagcgcctcacagtcgcgcg	SASASASAASASASASASAS
8636	8637	8657	20	22.742	46.585	181.0452	gagcgcctcacagtcgcgcg	SASASAASASASASASAS
8638	8639	8657	18	22.596	46.058	211.3387	gcgcctcacagtcgcgcg	SASAASASASASASAS
8639	8640	8658	18	22.903	26.212	152.9743	cgctcacagtcgcgcgt	ASSASASASASASASA
8640	8641	8657	16	22.716	39.895	185.9997	gcctcacagtcgcgcg	SAASASASASASASAS
8641	8642	8658	16	22.543	29.313	223.5817	cctcacagtcgcgcgt	SASASASASASASASA
8642	8643	8659	16	22.675	39.990	194.3437	ctcacagtcgcgcgtt	ASASASASASASASAS
8643	8644	8660	16	22.430	29.941	252.6680	tcacaagtcgcgcgttc	SASASASASASASASA
8644	8645	8661	16	22.725	39.097	184.3317	cacaagtcgcgcgttc	ASASASASASASASAS
8645	8646	8662	16	22.423	29.834	254.6942	acaagtcgcgcgttct	SASASASASASASASA
8647	8648	8664	16	22.444	27.845	248.8339	aagtcgcgcgttctac	SASASASASASASASA
8649	8650	8666	16	22.543	27.097	223.7286	gtgcgcgttctactg	SASASASASASAASAS
14676	14677	14697	20	22.837	28.025	163.8871	gatatattttcacgcacgcc	ASASASASASASASAS
14678	14679	14697	18	22.487	31.433	237.7120	tatattttcacgcacgcc	ASASASASASASASAS
14679	14680	14696	16	22.642	36.766	201.2662	atattttcacgcacgc	SASASASASASASASA
14680	14681	14697	16	22.181	32.253	332.9764	tattttcacgcacgcc	ASASASASASASASAS
14682	14683	14699	16	22.119	32.252	357.1333	ttttcacgcacgcoga	ASASASASASASASAS
14684	14685	14701	16	22.588	29.888	213.1408	ttcacgcacgcgcgagc	ASASASASASSASASA
14686	14687	14703	16	22.552	31.210	221.3909	cacgcacgcgcgcgcg	ASASASASSASASASA
18946	18947	18967	20	22.805	53.618	169.3967	tttgtgtgtccggacacac	SASASASASASASASASA
18948	18949	18967	18	22.673	53.535	194.5953	tttgtgtgtccggacacac	SASASASASASASASASA
18949	18950	18968	18	22.916	55.576	151.0613	tgtgtgtccggacacaca	ASASASASASASASAS
18950	18951	18967	16	22.548	52.759	222.4103	gtgtgtccggacacac	SASASASASASASASA

- 2) Оказалось, что в моем файле нет квадруплексов

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>
Requirement already satisfied: biopython in /usr/local/lib/python3.10/dist-packages (1.81)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from biopython) (1.22.4)
len(PQS) = 0

Поэтому проведу анализ для chr19 Bos Taurus (в дальнейшем, все что связано с Z-ДНК будет происходить с геном вируса Эбола, а то что связано с квадруплексами – с 19 хромосомой Bos Taurus)

Число квадруплексов : 1544

```
PQS = [[x.start(), x.end(), x.group(0)] for x in re.finditer(reg, seq)]
```

```
print('len(PQS) = {}'.format(len(PQS)))
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: biopython in /usr/local/lib/python3.10/dist-packages (1.81)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from biopython) (1.22.4)
len(PQS) = 1544
```

3) Скачала аннотацию с сайта

https://www.ncbi.nlm.nih.gov/genome/gdv/browser/genome/?id=GCF_002263795.2

И загрузила ее в colab

Затем при помощи кода (весь можно будет посмотреть в notebook) ответила на вопрос.

Результат:

```
print('----- Для Z-DNA -----')
print('Гены: {}'.format(inner_count))
print('Межгенное пространство: {}'.format(outer_count))
print('Пересечение: {}'.format(between_count))
```

```
----- Для Z-DNA -----
Гены: 0
Межгенное пространство: 24
Пересечение: 0
```

Напомню, что это посчитано для вируса Эбола, в связи с этим маленькие значения.

Опционально прилагаю результат поиска для 19 хромосомы Bos Taurus:

```
print('----- Для Z-DNA -----')
print('Гены: {}'.format(inner_count))
print('Межгенное пространство: {}'.format(outer_count))
print('Пересечение: {}'.format(between_count))
```

```
----- Для Z-DNA -----
Гены: 25
Межгенное пространство: 727
Пересечение: 0
```

Итак, теперь квадруплексы. Ура-ура!! Тут всё на месте.

```
print('----- Для квадруплексов -----')
print('Гены: {}'.format(inner_count))
print('Межгенное пространство: {}'.format(outer_count))
print('Пересечение: {}'.format(between_count))
```

```
----- Для квадруплексов -----
Гены: 842
Межгенное пространство: 673
Пересечение: 29
```

4) Не нашлось ни одного гена у вируса Эбола, в промотеры которых попало Z-ДНК

For gene LOC101906311 we have 0 Z-DNA inside
For gene GPR142 we have 0 Z-DNA inside
For gene BTBD17 we have 0 Z-DNA inside
For gene LOC112442753 we have 0 Z-DNA inside
For gene LOC112442727 we have 0 Z-DNA inside
For gene LOC112442728 we have 0 Z-DNA inside
For gene LOC112442724 we have 0 Z-DNA inside
For gene LOC112442729 we have 0 Z-DNA inside
For gene LOC112442726 we have 0 Z-DNA inside
For gene LOC112442725 we have 0 Z-DNA inside
For gene KIF19 we have 0 Z-DNA inside
For gene DNAI2 we have 0 Z-DNA inside
For gene TTYH2 we have 0 Z-DNA inside
For gene RPL38 we have 0 Z-DNA inside
For gene LOC101907074 we have 0 Z-DNA inside
For gene LOC112442754 we have 0 Z-DNA inside
For gene SDK2 we have 0 Z-DNA inside
For gene LOC112442730 we have 0 Z-DNA inside
For gene CDC42EP4 we have 0 Z-DNA inside
For gene CPSF4L we have 0 Z-DNA inside
For gene C19H17orf80 we have 0 Z-DNA inside
For gene FAM104A we have 0 Z-DNA inside
For gene COG1 we have 0 Z-DNA inside
For gene SLC39A11 we have 0 Z-DNA inside
For gene SSTR2 we have 0 Z-DNA inside
For gene LOC101907523 we have 0 Z-DNA inside
For gene LOC112442732 we have 0 Z-DNA inside

Однако для коровы встречается: LOC107131483, например.


5) Теперь аналогичная операция для квадруплексов.

Приведу список «ненулевых» генов

For gene LOC112442606 we have 1 QUAD`s inside
For gene SLC13A2 we have 1 QUAD`s inside
For gene LOC527796 we have 1 QUAD`s inside
For gene LOC112442625 we have 1 QUAD`s inside
For gene TEK11 we have 1 QUAD`s inside
For gene LOC613988 we have 1 QUAD`s inside
For gene TRNAK-UUU we have 1 QUAD`s inside
For gene SPAG7 we have 1 QUAD`s inside
For gene ELP5 we have 1 QUAD`s inside
For gene LOC104975041 we have 1 QUAD`s inside
For gene LOC101907886 we have 1 QUAD`s inside
For gene LOC101902768 we have 1 QUAD`s inside
For gene MAPK7 we have 1 QUAD`s inside
For gene SP2 we have 1 QUAD`s inside
For gene TRNAW-CCA we have 1 QUAD`s inside
For gene MIR4286-1 we have 1 QUAD`s inside
For gene ZPBP2 we have 1 QUAD`s inside
For gene KRT27 we have 1 QUAD`s inside

For gene TTC25 we have 1 QUAD`s inside
 For gene NKIRAS2 we have 1 QUAD`s inside
 For gene CCDC43 we have 1 QUAD`s inside
 For gene GJC1 we have 1 QUAD`s inside
 For gene CYTH1 we have 1 QUAD`s inside
 For gene MIR2349 we have 1 QUAD`s inside
 For gene LOC104975119 we have 1 QUAD`s inside
 For gene MGAT5B we have 1 QUAD`s inside
 For gene METTL23 we have 1 QUAD`s inside
 For gene LOC112442779 we have 1 QUAD`s inside
 For gene MRPS7 we have 1 QUAD`s inside
 For gene LOC112442717 we have 1 QUAD`s inside
 For gene SLC25A19 we have 1 QUAD`s inside
 For gene LOC104975137 we have 2 QUAD`s inside
 For gene WBP2 we have 1 QUAD`s inside
 For gene LOC107131544 we have 1 QUAD`s inside
 For gene EXOC7 we have 1 QUAD`s inside

6) Скрин анализа + файл с названием task6-bos-taurus


Search Download Help My Data

The following proteins in *Bos taurus* appear to match your input. Please review the list, then click 'Continue' to proceed.

<- BACK
! MAPPING
CONTINUE ->

35 query items showing page 1 of 2 • first • previous • next • last

- 1) 'LOC112442606':
-- Sorry, STRING found no proteins by this name in *Bos taurus* --
- 2) 'SLC13A2':
☒ [SLC13A2](#) - Bos taurus solute carrier family 13 (sodium-dependent dicarboxylate transporter), member 2 ([SLC13A2](#)), mRNA
- 3) 'LOC527796':
-- Sorry, STRING found no proteins by this name in *Bos taurus* --
- 4) 'LOC112442625':
-- Sorry, STRING found no proteins by this name in *Bos taurus* --
- 5) 'TEKT1':
☒ [TEKT1](#) - Tektin-1; Structural component of ciliary and flagellar microtubules. Forms filamentous polymers in the walls of ciliary and flagellar microtubules (By similarity)
- 6) 'LOC613988':
-- Sorry, STRING found no proteins by this name in *Bos taurus* --
- 7) 'TRNAK-UUU':
-- Sorry, STRING found no proteins by this name in *Bos taurus* --
- 8) 'SPAG7':
☒ [SPAG7](#) - Bos taurus sperm associated antigen 7 ([SPAG7](#)), mRNA

