

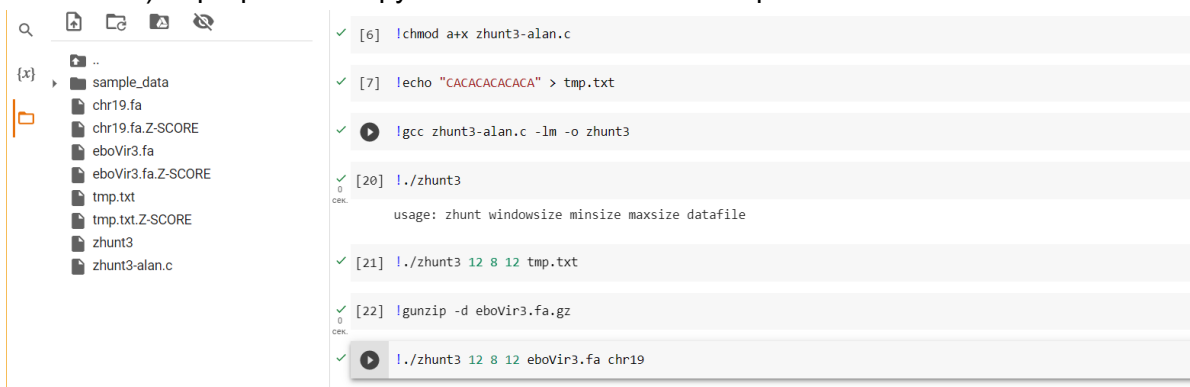
ОТЧЕТ

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
```

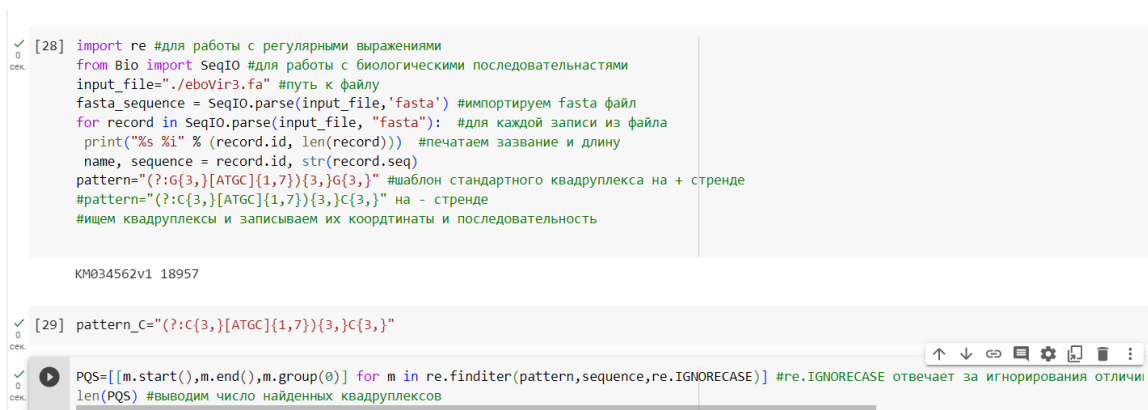
- 3) Данные обработаны и представлены в виде таблицы с полученными участками Z-ДНК.



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 table is as follows:

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 tgtcgattcaaaaaga 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 отличий
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	aggagcgctcacaagtcgcgcg	SASASASAASASASASASAS
8636	8637	8657	20	22.742	46.585	181.0452	gagcgctcacaagtcgcgcg	SASASAASASASASASASAS
8638	8639	8657	18	22.596	46.058	211.3387	gcgcctcacaagtcgcgcg	SASAASASASASASASAS
8639	8640	8658	18	22.903	26.212	152.9743	cgctcacaagtcgcgcgt	ASSASASASASASASASA
8640	8641	8657	16	22.716	39.895	185.9997	gcctcacaagtcgcgcg	SAASASASASASASAS
8641	8642	8658	16	22.543	29.313	223.5817	cctcacaagtcgcgcgt	SASASASASASASASA
8642	8643	8659	16	22.675	39.990	194.3437	ctcacaagtcgcgcgtt	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	ASASASASASASASASAS
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	gttgtgtccggacacac	SASASASASASASASA

- 2) Оказалось, что в моем файле нет квадруплексов

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

```
PQS_minus=[m.start(),m.end(),m.group(0)] for m in re.finditer(pattern_C,sequence,re.IGNORECASE)] #re.IGNORECASE отвечает за игнорировани
len(PQS_minus) #выводим число найденных квадруплексов
```

Поэтому проведу анализ для chr19 COW.

```
[10] #скачиваем 19 хромосому с UCSC
!wget https://hgdownload.soe.ucsc.edu/goldenPath/bosTau4/chromosomes/chr19.fa.gz
#развархивируем
!gunzip ./chr19.fa.gz
```

```
--2023-06-06 19:22:14-- https://hgdownload.soe.ucsc.edu/goldenPath/bosTau4/chromosomes/chr19.fa.gz
Resolving hgdownload.soe.ucsc.edu (hgdownload.soe.ucsc.edu)... 128.114.119.163
Connecting to hgdownload.soe.ucsc.edu (hgdownload.soe.ucsc.edu)|128.114.119.163|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20166379 (19M) [application/x-gzip]
Saving to: 'chr19.fa.gz'
```

```
chr19.fa.gz          100%[=====>] 19.23M  9.25MB/s   in 2.1s
```

```
2023-06-06 19:22:17 (9.25 MB/s) - 'chr19.fa.gz' saved [20166379/20166379]
```

Число квадруплексов для 19 хромосомы коровы: всего 9077

```
✓ [15] PQS=[ [m.start(),m.end(),m.group(0)] for m in re.finditer(pattern,sequence,re.IGNORECASE)] #re.IGNORECASE отвечает за игнорирования отличии
3 сек. len(PQS) #выводим число найденных квадруплексов
9077
```

```
✓ [16] PQS_minus=[ [m.start(),m.end(),m.group(0)] for m in re.finditer(pattern_C,sequence,re.IGNORECASE)] #re.IGNORECASE отвечает за игнорировани
2 сек. len(PQS_minus) #выводим число найденных квадруплексов
8935
```