II. Постройте позиционную весовую матрицу (PWM) для последовательности Козак человека и проверьте её.

1. Постройте PWM
	1. Выберите 100 (?) генов из генома   [не менее 30 и не более 100]
	2. Вырежьте фрагменты (7 нуклеотидов до старта трансляции ATG) + ATG + 3 нук. после ATG.  Итого фрагмент длины 13 нукл. Внимательно отнеситесь к тому, с какой цепочки скачиваете!
	3. Разделите данные на две части, меньшую обучение 40 генов (если 100) и тест 60 генов.
	4. Составьте из фрагментов из материала обучения «выравнивание» без гэпов (все ATG друг под другом)
	5. Постройте PWM (cм. презентацию)
2. Вычислите веса построенной PWM для тестовой выборки фрагментов.
*Вариант 1* в геноме SARS-CoV-2
\* положительный контроль – старт сайты ATG для генов SARS-CoV-2;- их 11, полипротеин и 10 поздних генов. Координата найдете в записи с геном в формате .gb (ссылка есть)
\* Отрицательный контроль – все остальные ATG в геноме (или часть из них), кроме отобранных. Для них нет никаких оснований ожидать высоко веса PWM для посл-ти Козак.
Дан геном SARS-CoV-2 и координаты всех ATG, см ссылки из задания.

*Вариант 2*\* положительный контроль множество тестовое генов человека

\* отрицательный отберите сами ATG, которые не являются стартами трансляции, т.е. лежат вне кодиующей последовательности или наоборот лежт внутри гена.

 3. Сравните распределения весов для положительного и отрицательного контроля и сделайте выводы.

Обязательно посмотрите в подсказки, это облегчит вам жизнь (MV).

**Подсказки**

1a Таблицы генов для выбора <https://genome.ucsc.edu/cgi-bin/hgTables>Предлагается скачанная табл с генами белков, отфильтрованная, но всё равно генов многовато.
В таблице координаты начала первого кодона ATG указаны как Thickstart  если на прямой цепи, и ThickEnd – если на обратной.

1b python
     \* Если библиотека Request не входит в вашу установку python, для упрощения кода запросов командой инсталлируйте эту библиотеку.
     python -m pip install requests
     [Requests is an elegant and simple HTTP library for Python, built for human beings. <https://docs.python-requests.org/en/latest/>]

             \* Вот код для скачивания одного фрагмента генома человека по заданным координатам:

Чтобы найти нужный нукл. участок из базы данных необходимо ввести следующую команду:

*"*[*https://rest.ensembl.org/sequence/region/human/{номер*](https://rest.ensembl.org/sequence/region/human/%7B%D0%BD%D0%BE%D0%BC%D0%B5%D1%80) *хромосомы}:{thickStart}..{thickEnd}:{ прямая или обратная цепь : 1, -1}?expand\_3prime={кол-во нуклеотидов после стоп -кодона};expand\_5prime={кол-во нуклеотидов перед ATG}"*

Например, для гена из таблицы найдем последовательность гена + 10 букв от старт-кодона:

*"*[*https://rest.ensembl.org/*](https://rest.ensembl.org/sequence/region/human/%7B%D0%BD%D0%BE%D0%BC%D0%B5%D1%80)*sequence/region/human/1:450739..451678:-1?expand\_3prime=0;expand\_5prime=10"*

Внизу приведен пример кода, для автоматического скачивания фрагмента генома:

import requests, sys
server = “[https://rest.ensembl.org”](https://rest.ensembl.org)ext = "/sequence/region/human/X:1000000..1000100:1?expand\_3prime=60;expand\_5prime=60"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-fasta"})
if not r.ok:
    r.raise\_for\_status()

    sys.exit()
print(r.text)

Координаты описываются в переменной ext. В примере геном человека, хромосома X, 1 = + цепочка (бывает -1) расширить координаты на 60 нукл с 3' и 5’ концов.
           У меня python 3.8.1 всё получилось

 Мария Волобуева (MV) проверила, что скрипт работает корректно, если задать корректные входные данные. Адрес, можно вопросы задавать (mashila6799@gmail.com). Она старшекурсница.

1b На странице <https://rest.ensembl.org/documentation/info/sequence_region>

найдёте как тот же пример выполнить с помощью bash команд  curl или wget.  Проверил – тоже работает. Там же есть описание параметров для скачивания.

2. Тестовая выборка
 (1) такие же фрагменты для других генов человека
 (2) 11 генов SARS-CoV-2 + не действующие ATG

Для проверки напишите программу, которая на вход получает PWM и набор последовательностей, в данном случае, фрагментов 7+3+3, на выходе – веса каждого из фрагментов относительно PWM. В этом задании я не программирование проверяю. Поэтому можно использовать программу написанную другим студентом, конечно с указанием автора.