МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science»

Слушатель Чувакова Любовь Николаевна

Москва, 2023 год.

# Содержание

Введение

1. Аналитическая часть

1.1. Постановка задачи

1.2. Описание используемых методов

1.2.1 Scanpy

1.2.2 Scrublet

1.2.3 PCA

1.2.4 Метод k-ближайших соседей

1.2.5 tSNE

1.2.6 Leiden

1.2.7 UMAP

1.2.8 DBSCAN

1.2.9 RANDOM FOREST

1.2.10 Нейронная сеть

2. Практическая часть

2.1. Предобработка данных

2.1.1 Удаление пустых клеток

2.1.2 Удаление клеток с митохондриальными генами

2.1.3 Удаление дублетов

2.1.4 Нормализация и логарифмирование данных

2.2 Анализ данных

2.2.1 Снижение размерности – Метод главных компонент

2.2.2 Метод k-ближайших соседей

2.2.3 tSNE

2.2.4 Leiden

2.2.5 Нейронная сеть

2.3. Разработка приложения

2.4. Создание удаленного репозитория

Заключение

# Введение

Тема данной работы - Кластеризация клеток периферической крови от здорового донора после single-cell RNA sequencing.

Single-cell RNA sequencing (scRNA-seq) – относительно новый метод секвенирования РНК, который активно внедряется в научную практику. Этот метод позволяет получать данные от отдельных клеток различной этиологии, оценить, сколько и каких уже известных клеток было в образце, пути дифференциации клеток, а также возможные функции клеток в их микроокружении. Помимо этого, scRNA-seq позволяет выявить, выделить и проанализировать новые популяции клеток. Чтобы выделить популяции клеток, необходимо их кластеризовать, что уже успешно реализуется с помощью методов машинного обучения.

В данной работе использован датасет, полученный в результате scRNA-seq образца периферической крови, и после его картирования на референсный геном. Исходный датасет подвергнут препроцессингу, в виде удаления пустых значений (пустых капель), капель с погибшими клетками (с высоким содержанием митохондриальных генов), дуплетов, нормализации. Проведен кластерный анализ разными методами машинного обучения, для определения количества кластеров популяций клеток крови, с возможностью дальнейшего более глубокого анализа какого-либо кластера.

1. **Аналитическая часть.**

Полный цикл scRNA-Seq, от пробоподготовки до конечной кластеризации обычно включает следующие этапы: выделение отдельной клетки, экстракция и амплификация нуклеиновых кислот, подготовка библиотеки секвенирования, секвенирование и анализ биоинформатических данных (рис.1)

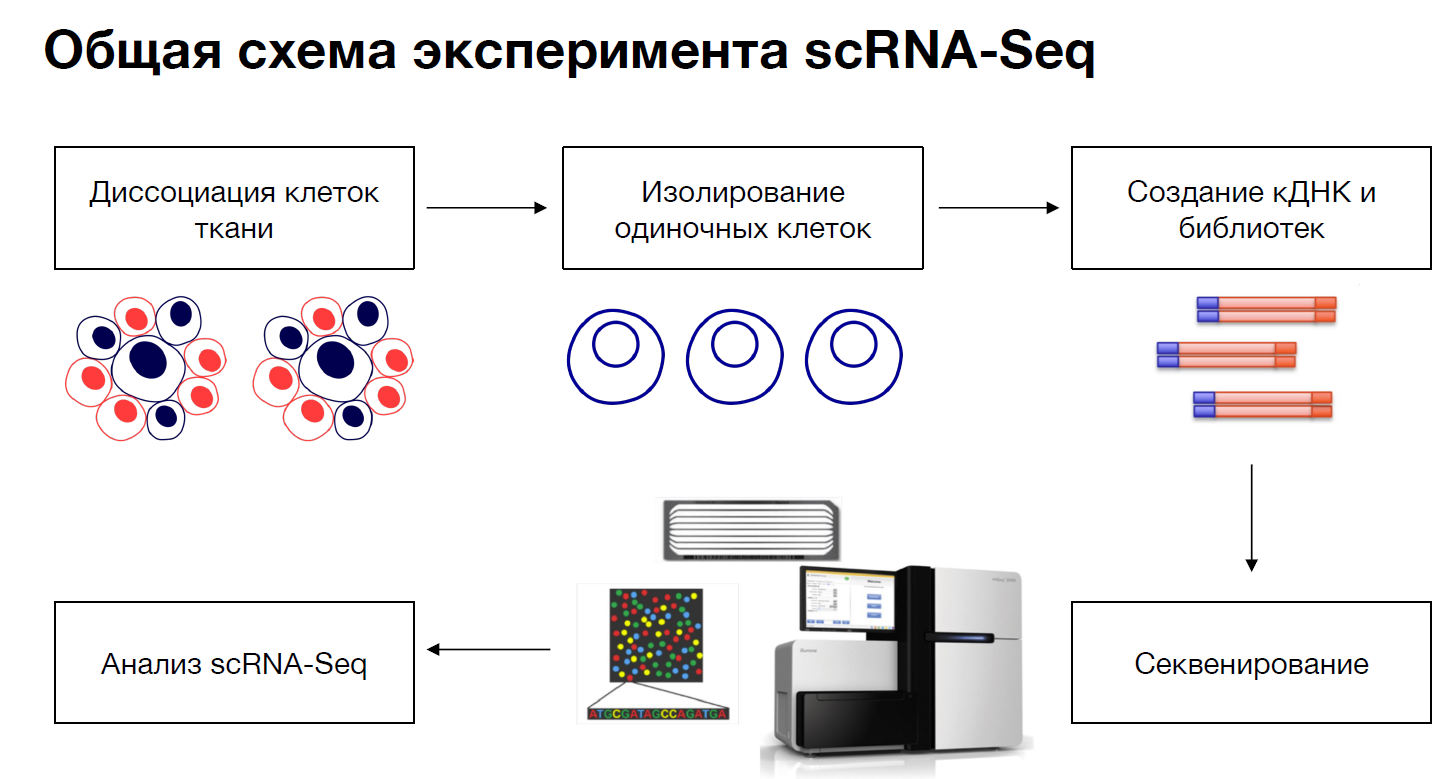
****

Рис 1. Общая схема эксперимента

Первым шагом метода является разделение и инкапсуляция каждой клетки и подготовка библиотеки. Клетки инкапсулируются в гелевые шарики в эмульсии (GEM) (рис. 2).

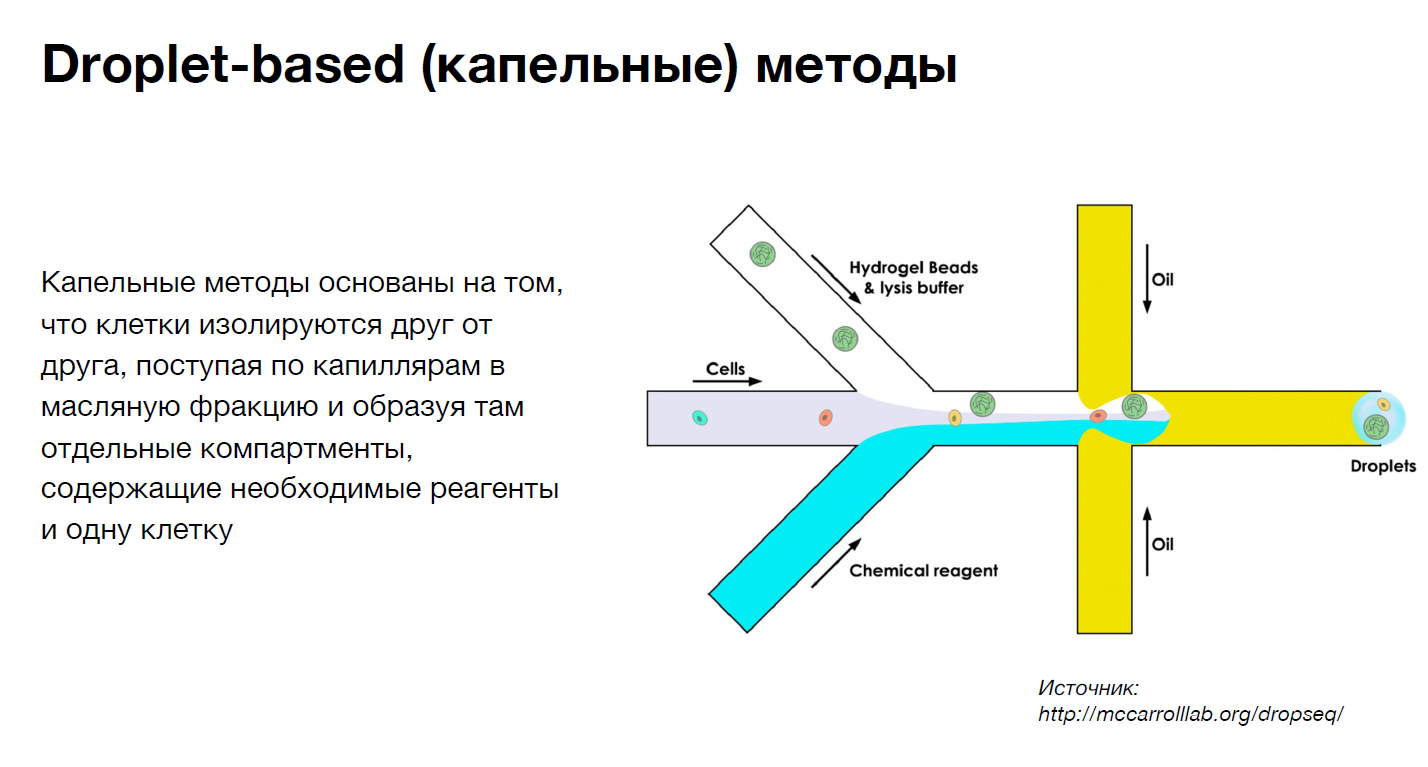
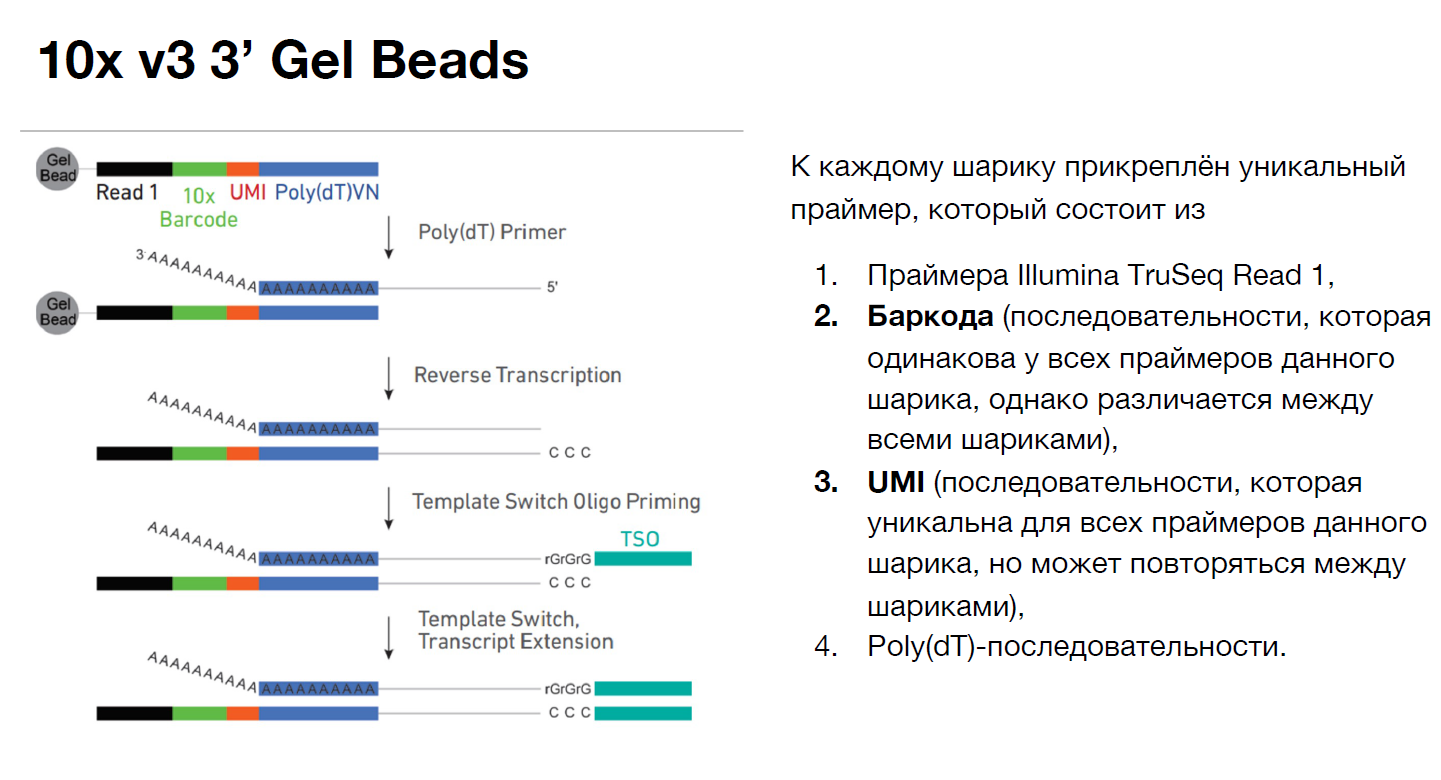
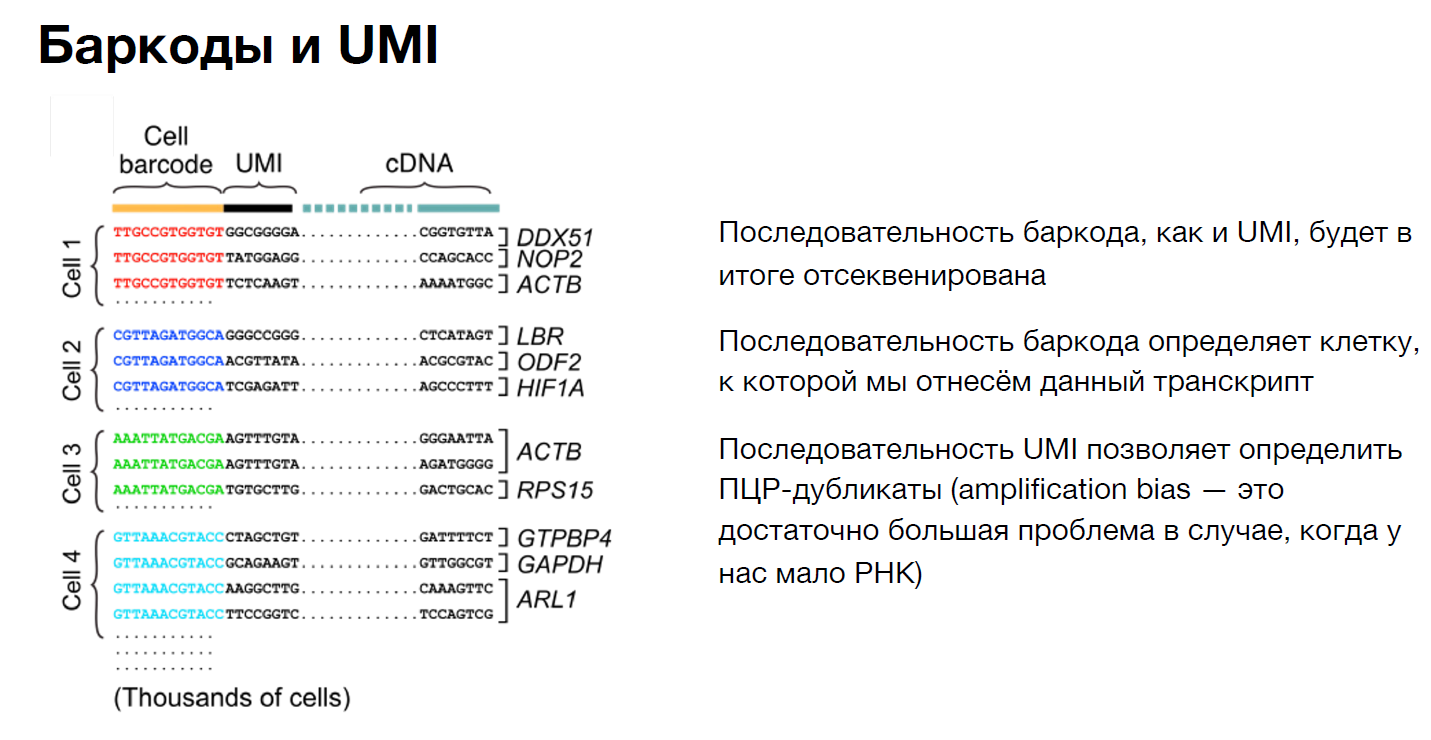
****

Рис. 2. Изоляция диссоциированных клеток в дроплеты

Каждый функциональный GEM содержит одну ячейку, одну гелевую бусину и реагенты RT. На гелевой бусине связываются олигнонуклеотиды, состоящие из 4 отдельных частей: праймер для ПЦР (необходим для секвенирования); 10X олигонуклеотиды со штрих-кодом; последовательность уникального молекулярного идентификатора (UMI); Последовательность PolydT (которая позволяет захватывать полиаденилированные молекулы мРНК) (рис. 3).

****

****

Внутри каждой реакционной везикулы GEM одна клетка лизируется и подвергается обратной транскрипции. кДНК из одной и той же клетки идентифицируются благодаря общему штрих-коду 10X. Кроме того, количество UMI выражает уровень экспрессии генов, и его анализ позволяет обнаруживать высоковариабельные гены. Эти данные часто используются либо для классификации клеточных фенотипов, либо для идентификации новых субпопуляций.

Завершающим этапом платформы является секвенирование. Созданные библиотеки можно напрямую использовать для секвенирования всего транскриптома одной клетки или рабочих процессов целевого секвенирования.

В дальнейшем осуществляется непосредственно биоинформатический анализ данных. Из секвенатора выгружаются данные в формате .fastq. Выравнивание и подсчет экспрессии генов на клетку выполняется программой. Золотым стандартом является программа Cell Ranger. Ее достоинством является широкий спектр функций, начиная от автоматичского определения координат координаты баркода / UMI в прочтениях, работает с разными модификациями scRNA-Seq, с помощью флагов можно провести препроцессинг данных. Однако, он очень требовательный к ресурсам (1 Тб дискового пространства, 128 Гб RAM, 16 ядер) и очень долго работает (один образец может рассчитываться 12 часов).

В простейшем случае аутпут содержит 4 файла:

1. raw\_feature\_bc\_matrix.tar.gz — матрица со всеми “клетками” из датасета (в дальнейшем и будет анализироваться)

a. barcodes.tsv.gz — названия клеток (баркоды)

b. features.tsv.gz — названия и id генов

c. matrix.mtx.gz — непосредственно матрица экспрессии в sparce-виде

2. filtered\_feature\_bc\_matrix.tar.gz — то же, что и пункт 1, только с уже

отфильтрованными клетками (Cell Ranger фильтрует очень неплохо)

a. barcodes.tsv.gz

b. features.tsv.gz

c. matrix.mtx.gz

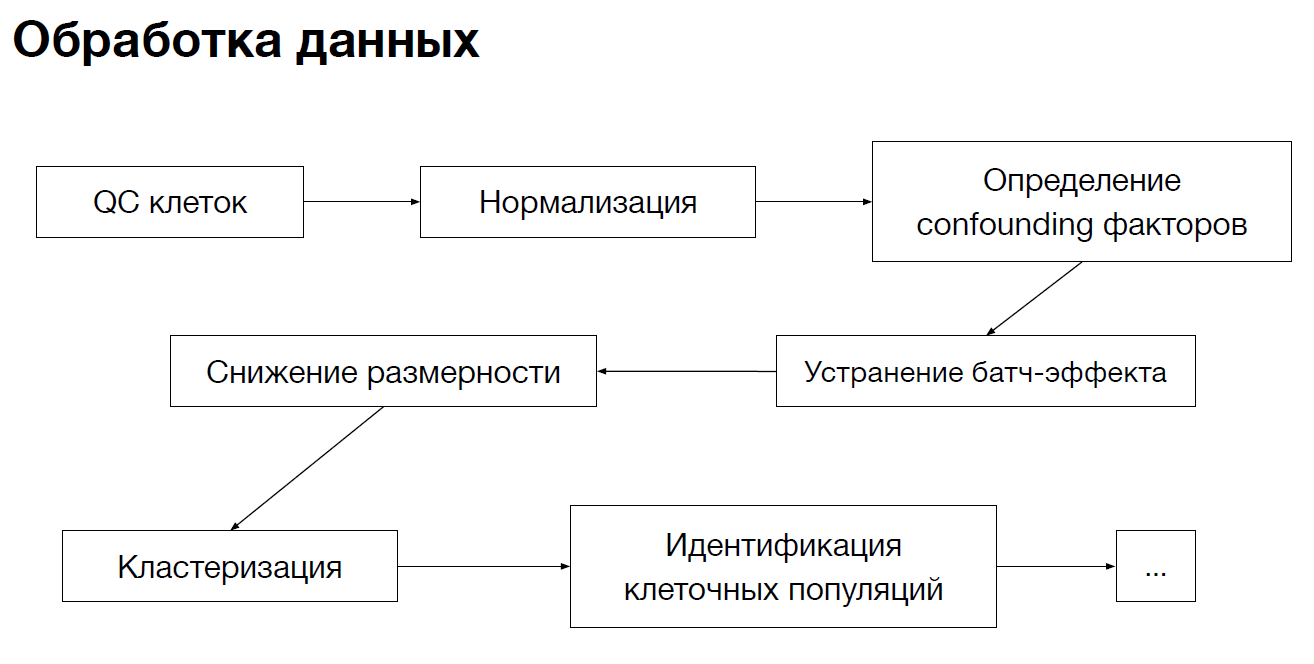
3. metrics\_summary.csv — таблица с основными метриками

4. web\_summary.html — графический веб-отчёт о качестве выравнивания и т. п.

* 1. Постановка задачи

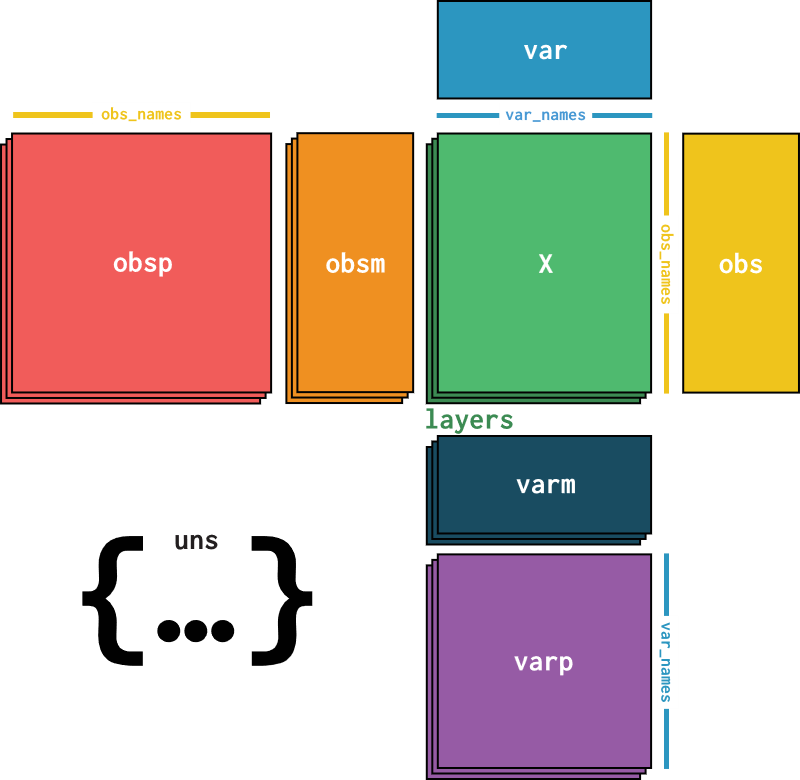
В папке outs содержатся файлы после отработки CellRanger. Датасет raw\_feature\_bc\_matrix.tar.gz находится на google disk, так как Githab не позволяет подгружать большие датасеты с весом, более 25 мб непосредственно в репозиторий и более 100 мб через командную строку.

Для того, чтобы кластеризовать данные после scRNA-Seq и в дальнейшем выбрать и проанализировать интересующие кластеры клеток периферической крови, необходимо сначала провести препроцессинг данных, который включает в себя QC клеток (удаление пустых капель (аналог удаления строк с NA), капель с погибшими клетками (где большое количество митохондриальных генов), дуплетов, нормализацию данных, возможные дополнительные этапы и затем непосредственно кластеризация клеточных популяций (рис.4).

****

# Данные представлены в формате AnnData – Annotated data – это так называемый контейнерный формат данных, где данные взаимосвязаны между собой, но представлены в нескольких таблицах (рис.5).

***Class*anndata.AnnData(*X=None*, *obs=None*, *var=None*, *uns=None*, *obsm=None*, *varm=None*, *layers=None*, *raw=None*, *dtype=None*, *shape=None*, *filename=None*, *filemode=None*, *asview=False*, *\**, *obsp=None*, *varp=None*, *oidx=None*, *vidx=None*)**



[AnnData](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.html#anndata.AnnData) хранит в себе матрицу [X](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.X.html#anndata.AnnData.X) вместе с аннотациями исследований [obs](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.obs.html#anndata.AnnData.obs) ([obsm](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.obsm.html#anndata.AnnData.obsm), [obsp](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.obsp.html#anndata.AnnData.obsp)), вариативными данными [var](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.var.html#anndata.AnnData.var) ([varm](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.varm.html#anndata.AnnData.varm), [varp](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.varp.html#anndata.AnnData.varp)), и неструктурированными аннотациями [uns](https://anndata.readthedocs.io/en/latest/generated/anndata.AnnData.uns.html#anndata.AnnData.uns) (https://anndata.readthedocs.io/en/latest/#). Данный формат возможно перевести в data frame, однако часть данных будет потеряна, что не всегда может положительно сказаться на кластерном анализе.

В исходном датасете очень много клеток (больше трёх миллионов).

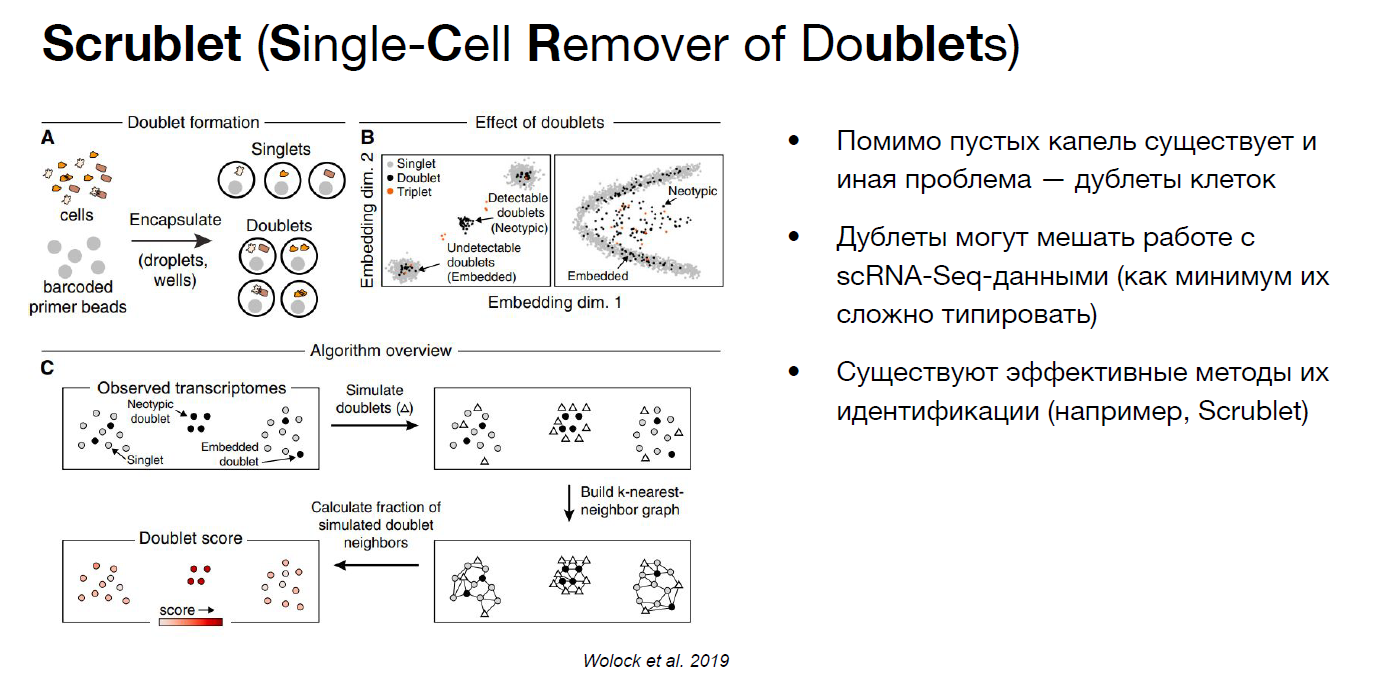
Основной объект AnnData — это матрица, в которой находятся данные по числу UMI каждого гена на каждую клетку. В нашем случае эта матрица имеет размерность число клеток × число генов, то есть 3069478 × 36601. Эта матрица пока находится в sparse-виде. Это такой формат, который подразумевает, что нулей в матрице сильно больше, чем ненулевых элементов — тогда в явном виде можно хранить только координаты ненулевых элементов, это сильно экономит память.

* 1. Описание используемых методов
     1. Scanpy – Single-Cell Analysis in Python

Scanpy — это масштабируемый инструментарий для анализа данных об экспрессии генов в отдельных клетках, созданный совместно с anndata. Он включает в себя предварительную обработку, визуализацию, кластеризацию, вывод траектории и тестирование дифференциальных выражений. Реализация на основе Python эффективно работает с наборами данных, содержащими более миллиона ячеек (<https://scanpy-tutorials.readthedocs.io/en/latest/index.html>**)**

1.2.2. Scrublet

Scrublet используется для удаления капель, где оказалось 2 и более клеток. С такой ячейки приходит смешанная информация, которую затем анализировать не представляется возможным, поэтому данной утилитой строки с дублетами удаляются. Подход достаточно простой, сначала на строках генерируются фантомные дублеты и с ними сравниваются другие строки, и где оказывается похожее распределение по баркодам и UMI, те строки помечаются как дублеты и в дальнейшем из датасета удаляются.

****

1.2.3. PCA

Метод главных компонент (PCA) - один из основных способов уменьшить [размерность](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%BC%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%B0) данных, потеряв наименьшее количество [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), а также снизить шум в данных, который также может помешать адекватной кластеризации данных.

1.2.4 Метод k-ближайших соседей

Еще один метод классификации, который адаптирован для регрессии – метод k-ближайших соседей (k Nearest Neighbors). Суть метода проста: посмотри на соседей вокруг, какие из них преобладают, таковым ты и являешься.

Для реализации метода необходима метрика расстояния между объектами. Используется, например, эвклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных.

1.2.5 tSNE

tSNE - это алгоритм [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для [визуализации](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), является техникой [нелинейного снижения размерности](https://ru.wikipedia.org/w/index.php?title=%D0%9D%D0%B5%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%BE%D0%B5_%D1%81%D0%BD%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D0%B8&action=edit&redlink=1), хорошо подходящей для вложения данных высокой размерности для визуализации в пространство низкой размерности (двух- или трехмерное). В частности, метод моделирует каждый объект высокой размерности двух- или трёхмерной точкой таким образом, что похожие объекты моделируются близко расположенными точками, а непохожие точки моделируются с большой вероятностью точками, далеко друг от друга отстоящими.

1.2.6 Leiden

Алгоритм Лейдена использует идею ускорения локального перемещения узлов и идею перемещения узлов к случайным соседям (https://www.nature.com/articles/s41598-019-41695-z)

1.2.7. UMAP

Uniform Manifold Approximation and Projection (UMAP) — алгоритм [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), выполняющий [нелинейное снижение размерности](https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction). похожий на [t-SNE](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%BE%D1%85%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%BE%D1%81%D0%B5%D0%B4%D0%B5%D0%B9_%D1%81_t-%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%D0%BC), но с более сильным математическим обоснованием.

При снижении размерности UMAP сначала выполняет построение [взвешенного графа](https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D0%BE%D1%81%D1%81%D0%B0%D1%80%D0%B8%D0%B9_%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D0%B8_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2#%D0%B2%D0%B7%D0%B2%D0%B5%D1%88%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84), соединяя ребрами только те объекты, которые являются ближайшими соседями. Множество из ребер графа — это [нечёткое множество](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D1%87%D1%91%D1%82%D0%BA%D0%BE%D0%B5_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE) с [функцией принадлежности](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BD%D0%B0%D0%B4%D0%BB%D0%B5%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%B8), она определяется как вероятность существования ребра между двумя вершинами. Затем алгоритм создает граф в низкоразмерном пространстве и приближает его к исходному.

1.2.8 DBSCAN

Основанная на плотности пространственная кластеризация для приложений с шумами ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Density-based spatial clustering of applications with noise, DBSCAN) — это алгоритм [кластеризации данных](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), основанной на плотности — если дан набор точек в некотором пространстве, алгоритм группирует вместе точки, которые тесно расположены (точки со многими близкими соседями), помечая как выбросы точки, которые находятся одиноко в областях с малой плотностью (ближайшие соседи которых лежат далеко).

1.2.9 Случайный лес (RandomForest)

Случайный лес (RandomForest) - представитель ансамблевых методов.

Для определения входных данных каждому дереву используется метод случайных подпространств. Базовые алгоритмы обучаются на различных подмножествах признаков, которые выделяются случайным образом.

Преимущества случайного леса:

* высокая точность предсказания;
* редко переобучается;
* практически не чувствителен к выбросам в данных;
* одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки, данные с большим числом признаков;
* высокая параллелизуемость и масштабируемость.

Из недостатков можно отметить, что его построение занимает больше времени. Так же теряется интерпретируемость.

1.2.10 Нейронная сеть

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Структура нейронной сети пришла в мир программирования из биологии. Вычислительная единица нейронной сети — нейрон или персептрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа.

Смещение – это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения.

Так же у нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть. Примеры активационных функций: relu, сигмоида.

У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

* входной слой — его размер соответствует входным параметрам;
* скрытые слои — их количество и размерность определяем специалист;
* выходной слой — его размер соответствует выходным параметрам.

Прямое распространение – это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением.

Прогнозируемое значение сравниваем с фактическим с помощью функции потери. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась.

Для обновления весов в модели используются различные оптимизаторы.

Количество эпох показывает, сколько раз выполнялся проход для всех примеров обучения.

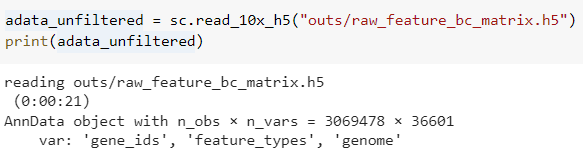
Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении.

2. Практическая часть

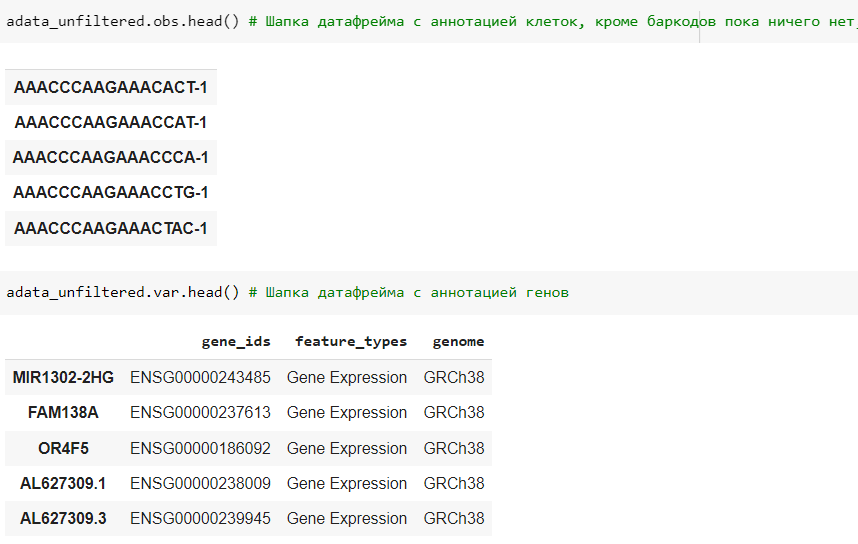
2.1. Предобработка данных

В исходном датасете очень много клеток (больше трёх миллионов).

Матрица имеет размерность число клеток × число генов, то есть 3069478 × 36601.

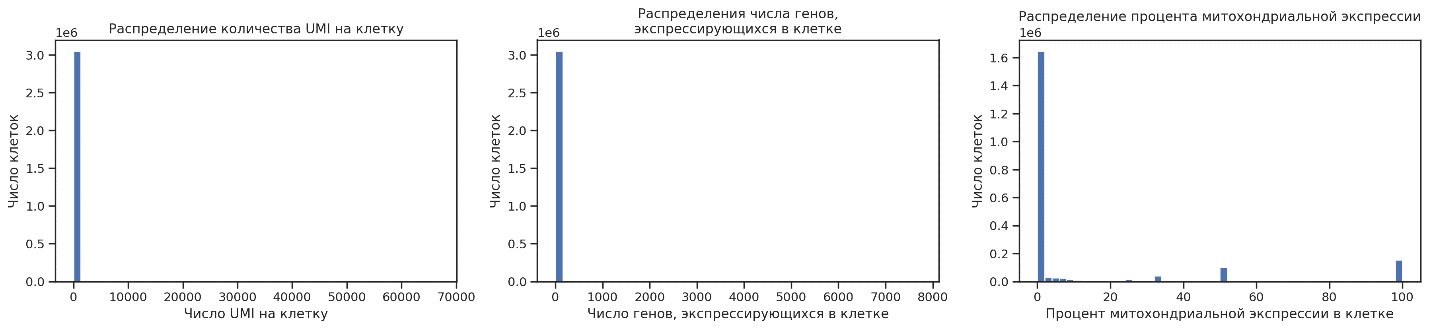


Проверяем, что разные обьекты в AnnData содержат информацию



Строим графики:

1. отрисовываем распределение общего числа UMI на клетку,
2. отрисовываем распределение числа генов по клеткам и
3. отрисовываем распределение митохондриальной экспрессии.



2.1.1 Удаление пустых клеток

При визуализации исходных данных видно, что подавляющее большинство «клеток» имеют околонулевую экспрессию. Фильтруем клетки, оставляем только те, что экспрессируют ≥ 100 UMI на клетку, то есть являются реальными клетками с мРНК, а не загрязнением внешней РНК при пробоподготовке.



И снова строим графики



Картина сильно улучшилась. Однако мы до сих пор видим сильный пик в начале. Для того, чтобы от него избавиться, можно просто вручную посмотреть минимум между двумя пиками (например, по числу генов) — здесь он находится где-то на 500 генах.

2.1.2 Удаление погибших клеток с митохондриальными генами.

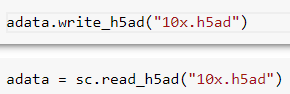
Если клетки погибли до или во время диссоциации ткани, то в таких клетках будет огромное количество дифференциально экспрессных генов митохондрий, которые своим фоном практически закрывают экспрессию мРНК клетки, что затрудняет процесс определения принадлежности клетки к какому-либо кластеру, поэтому от таких строк с клетками, тоже необходимо избавляться.



И также визуализируем полученные данные препроцессинга

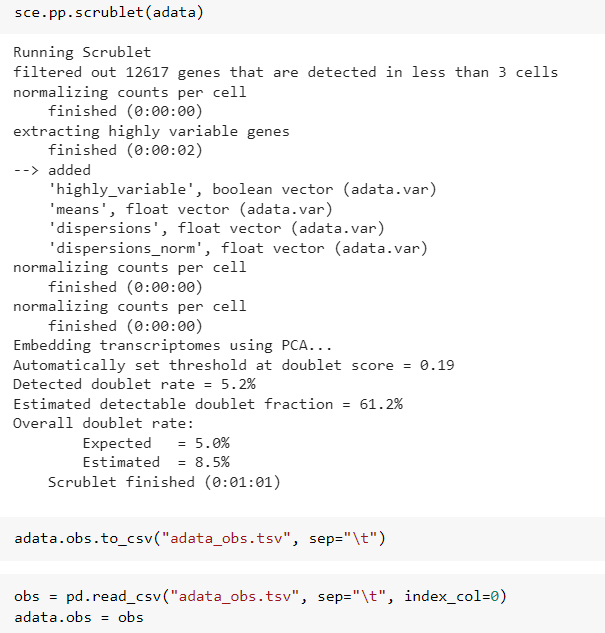


Записываем в формат AnnData наши данные, чтобы была возможность их в любой момент прочитать

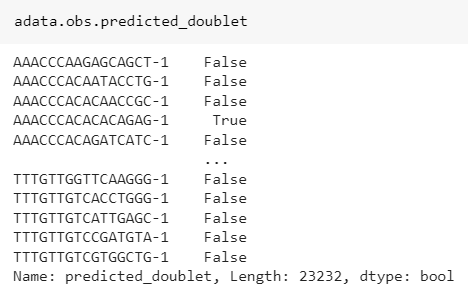


2.1.3 Удаление дублетов

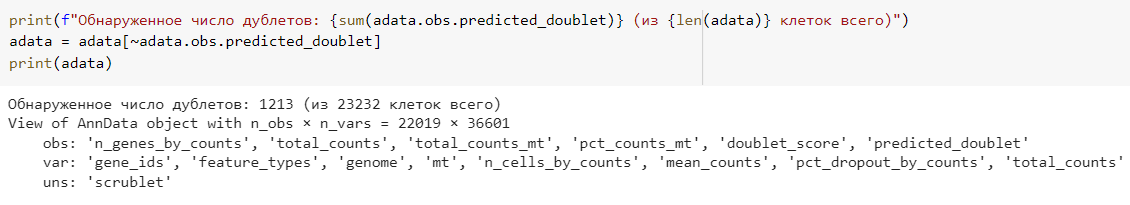
Помимо пустых капель у нас существует и другая проблема — дублеты из нескольких клеток. Для того, чтобы их выявить и удалить, мы используем пакет scrublet.



Теперь в adata.obs у нас появилась дополнительная колонка: predicted\_doublets



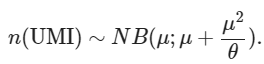
Теперь можно удалить дублеты (1213 дублетов из 23232 клеток)



2.1.4 Нормализация и логарифмирование данных

Наши данные из-за особенностей процедуры имеют слишком высокую техническую дисперсию (график не показан).

Вероятностная модель, которая описывает распределение числа UMI какого-то гена на клетку, представляет из себя обратное биномиальное распределение

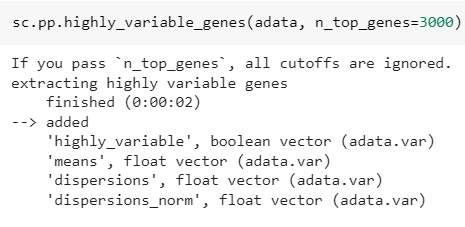


Для контроля над дисперсией можно использовать несколько различных подходов. Первый (самый ранний) проще всего — мы сначала нормируем количество каунтов каждой клетки так, чтобы это значение было равно какой-то константе (сейчас, как правило, это число — это медианное число каунтов в клетках датасета), а потом логарифмируем отнормированные каунты (а точнее берём функцию log(x + 1), чтобы избежать проблемы с нулями и иметь только положительные значения в результате). Такой метод в библиотеке Seurat называется LogNormalize.

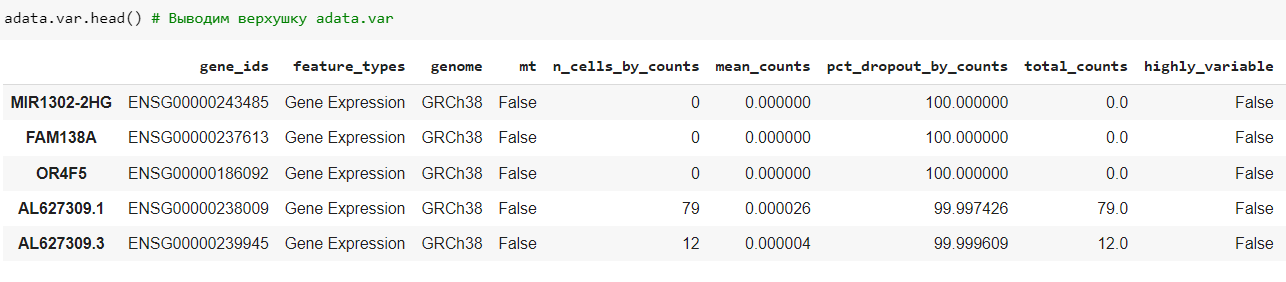


2.2 Анализ данных

Подготовка к кластеризации и снижению размерности. Выделяем самые высоко вариабельные гены. Это необходимо, чтобы амплифицировать сигнал, которая у нас есть в нашем датасете.



Теперь в описании adata.var есть отдельное поле, которое сообщает нам, является ли ген высоко вариабельным.

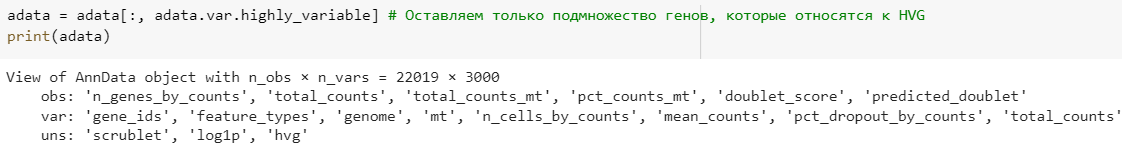


Сохраняем исходные данные в отдельный объект, чтобы к нему всегда можно было вернуться

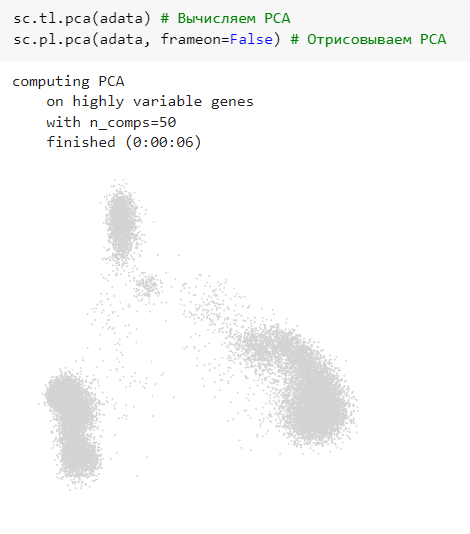


2.2.1 Снижение размерности – Метод главных компонент

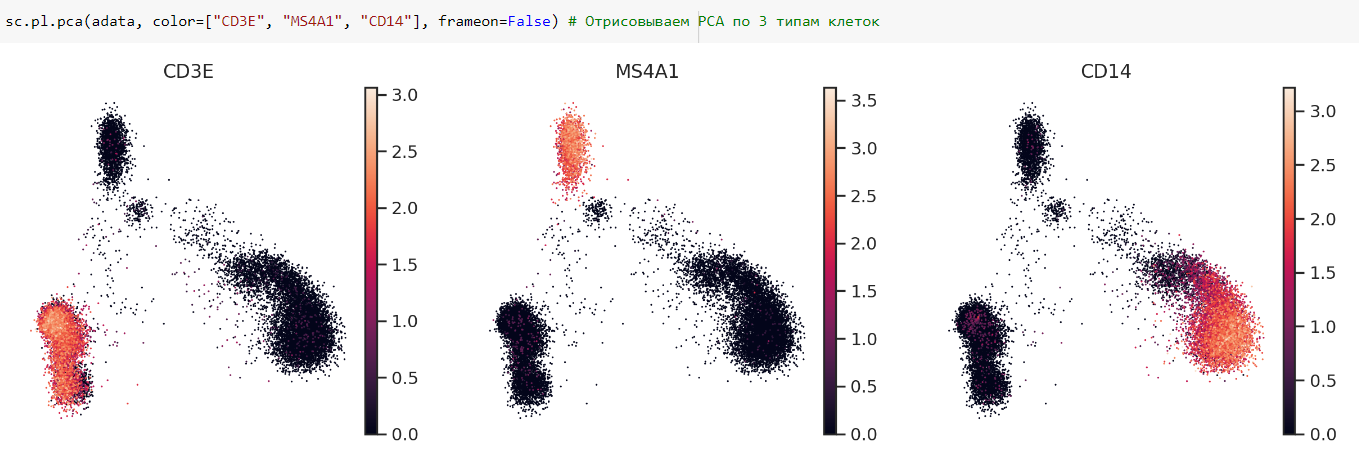
Дальше оставляем в нашем объекте только гены, которые мы отнесли к высоко вариабельным (HVG), то есть матрица становится 22019х3000



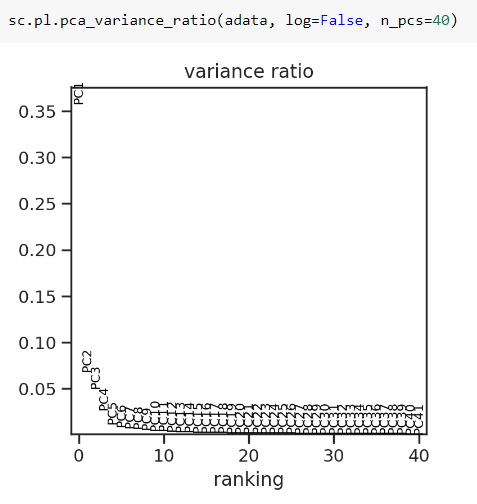
Проводим анализ методом главных компонент. Это позволит получить первичное представление о том, как данные распределены в пространстве, потребуется дальше, потому что первые N компонент PCA сохраняют максимальную биологическую значимость и в это же время устраняют шумы.



Часть клеток можно визуализировать, если знать маркерные гены и уровень их экспрессии:



Пока это далеко от красивой картинки, на которой отчётливо видны различные клеточные популяции. Но это нормально, потому что кластеризацию и анализ популяций мы будем проводить на UMAP. UMAP обычно строится на каком-то количестве первых компонент. Для того, чтобы их подобрать, нарисуем ElbowPlot.

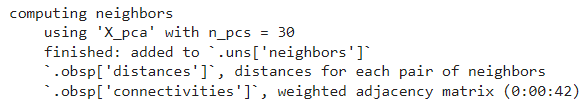


Теперь приступим к снижению размерности при помощи UMAP и t-SNE. Параметры для запуска UMAP и t-SNE мы возьмём некоторые дефолтные

2.2.2 Метод k-ближайших соседей

Вычисление графа k-ближайших соседей является важным этапом перед кластеризацией методами tSNE, UMAP и алгоритмом Leiden.





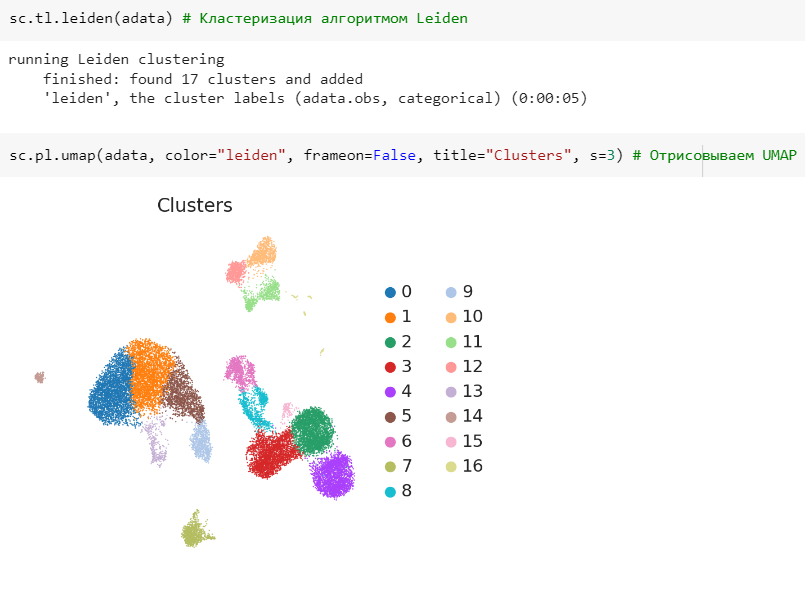
2.2.3 tSNE





2.2.4 Leiden кластеризация

Некоторым стандартом кластеризации является алгоритм leiden

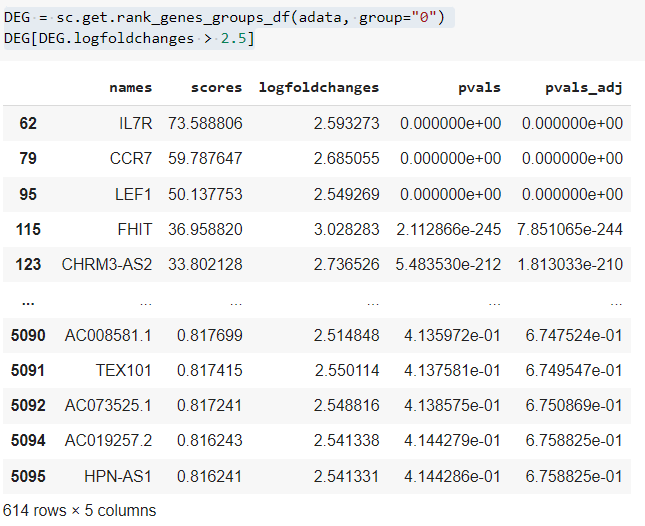


Возможное продолжение работы:

В дальнейшем необходимо понять, какой кластер соответствует какому типу клеток. Для этого мы можем:

1. определить, какие гены дифференциально экспрессированны в данном кластере, а потом, исходя из этого, определить тип клетки;
2. посмотреть, какие заранее известные гены-маркеры в каком кластере экспрессируются, и исходя из этого определить тип клетки.

Ниже приведен пример фильтрации данных 0 кластера



2.2.5 Нейронная сеть (в работе)

2.3. Разработка приложения

Разработка приложения, которое будет проводить весь необходимый анализ, представлена в виде файла app.py, однако, так и не была реализована на платформе. Причин несколько: 1) при запуске CellRanger возможно с помощью флагов прописать все необходимые этапы препроцессинга и получить уже готовые к кластеризации данные. 2) уже вышло приложение Azimuth, которое на вход принимает файл после картирования (в т.ч. после отработки CellRanger) в формате .h5, самостоятельно осуществляет препроцессинг, картирование и визуализацию данных (особенно касается часто секвенируемых фрагментов органов или тканей). 3) Если же транскриптом мало изучен, тогда есть смысл все этапы проводить вручную, чтобы контролировать все этапы подготовки данных.

## 2.7. Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/LyubovChuvakova/vkr\_work . На него были загружены результаты работы: исследовательский notebook, пояснительная записка, презентация и код приложения.

# Заключение

В ходе выполнения данной работы мы рассмотрели большую часть операций и задач, которые приходится выполнять специалисту по работе с данными.

Этот поток операций и задач включает:

* изучение теоретических методов анализа данных и машинного обучения;
* изучение основ предметной области, в которой решается задача;
* извлечение и транформацию данных. Здесь нам был предоставлен реальный набор данных, поэтому через трудности работы с разными источниками и парсингом данных мы еще не соприкоснулись;
* выполнение предобработки (препроцессинга) данных для обеспечения корректной работы моделей;
* построение аналитического решения. Это включает выбор алгоритма решения и модели, сравнение различных моделей;
* визуализация модели и оценка качества аналитического решения;

Я проделала максимум исследований, которые в моей компетенции как начинающего дата-сайентиста и ученого с биоинформатическим анализом данных, я применила большую часть знаний, полученных в ходе прохождения курса.

Дальнейшие возможные пути решения этой задачи могли бы быть:

* углубиться в изучение нейросетей, попробовать различные архитектуры, параметры обучения и т.д.;