**Data Engineer**
BostonGene
Benchling team

# Тестовое задание

## ОПИСАНИЕ ДАННЫХ

В Benchling мы регистрируем основные этапы подготовки тканей к секвенированию. В файле **benchling_data.xlsx** на вкладках представлены следующие этапы:

1) Specimen - регистрация ткани (WB - кровь, FPB - блок ткани, N - нормальная ткань, T - опухолевая ткань). Здесь генерируется имя образца для следующих этапов.
2) Plasma - регистрация плазмы из крови. Для некоторых проектов нужен промежуточный этап, такой как выделение плазмы из крови. Плазма всегда выделяется из нормальной ткани.
3) Extraction - экстракция ДНК и РНК из ткани или плазмы. Здесь генерируем D-id для ДНК и R-id для РНК.
4) Library - формирование библиотек из полученных экстрактов. Колонка Index - индекс последовательности нуклеотидов, которая позволяет секвенатору различать между собой отдельные библиотеки. Библиотека может быть сделана либо из РНК, либо из ДНК.
5) Sequencing Sample - формирование семплов для секвенирования. Для семпла нужна связка: ID библиотеки и имя Sequencing Run. Здесь есть важная колонка - Sample ID, которую мы генерируем с помощью скриптов. Она позволяет формировать нужную аннотацию образца для запуска дальнейших расчетов.

   **Правила формирования Sample ID:**

   a) Если семпл состоит из РНК, независимо от ткани:
   **RNASeq-tumor-<date>_<sequencer>_Sample_<Sample Num>**
   Пример: RNASeq-tumor-220623_NovaD_Sample_2

   b) Если семпл состоит из ДНК и ткань опухолевая:
   **WES-tumor-<date>_<sequencer>_Sample_<Sample Num>**
   Пример: WES-tumor-220622_NovaA_Sample_6

   c) Если семпл состоит из ДНК и ткань нормальная:
   **WES-normal-<date>_<sequencer>_Sample_<Sample Num>**
   Пример: WES-normal-220622_NovaA_Sample_2

## ЗАДАНИЕ

1) Написать функцию для заполнения колонки Sample ID во вкладке Sequencing Sample
2) Написать скрипт, который берет на входе benchling_data, заполняет Sample ID и выдает на выходе excel файлы по каждому Sequencing Run со следующими колонками:

| Sequencing Sample | | Library | Index | Specimen/Plasma | Clinical case/R&D case |
|---|---|---|---|---|---|
| 220623_NovaD_RNA_Sample_1 | | LIB000001 | IDX000001 | RS000136_T_FPB | RS000136 |
| 220623_NovaD_RNA_Sample_2 | | LIB000002 | IDX000288 | RS000136_T_FPB | RS000136 |
| 220623_NovaD_RNA_Sample_3 | | LIB000003 | IDX000289 | RS000137_T_FPB | RS000137 |
| … | | … | … | … | … |

   Каждый excel файл имеет имя рана (Sequencing Run), который был занесен в файл. Например, 220623_NovaD_RNA.xlsx

3) Обернуть все обращения к датафрейму в обработчик запросов. Разрешено делать не более 20 запросов в 30 секунд. То есть, если мы проходимся циклом по всем семплам, можно обрабатывать (читать или записывать) по 20 строк в 30 секунд.

4) Написать тест, который логирует (пишет в любом формате) сколько запросов было сделано к таблице и выдает Fail, если запросы были превышены или Success, если запросов в 30 секунд было не более 20.

5) Обработать возникновение ошибки **ConnectTimeoutError**, которая возникает в случае недоступности базы данных (нашей таблицы) в течение некоторого времени. При возникновении ошибки - подождать минуту и попробовать запрос снова. Эта ошибка может возникать как при чтении таблицы, так и при её обработке.
   Так как **ConnectTimeoutError**, не может возникнуть при обработке эксель файла, только при работе с реальной базой данных, в данном пункте будет проверяться логика, где была поставлена обработка, как представлен delay и повторный запуск.

Итогом задания будет скрипт на Python (можно jupyter notebook) по формированию отдельных файлов по каждому Sequencing Run, с обработкой количества запросов и времени ожидания при недоступности “базы данных”, а также отдельная тест-функция по измерению количества и времени запросов к таблице.

Успехов!