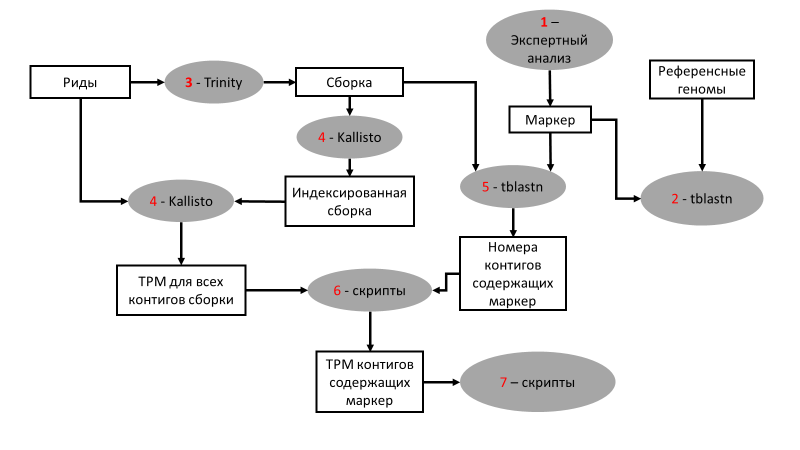
Структура всей технической части работы представлена на рисунке 1. Все скрипты, позволяющие связать различные этапы работы, а также скрипты, необходимые для последних этапов анализа, были написаны на языке программирования Python.



*Рис 1.* Схематическое представление этапов технической части работы. Для каждого этапа показано каким образом была проведена каждая техническая операция.

***Этап 1 – Анализ литературы***

Экспертный анализ позволил подобрать каждой функциональной группе 1 или несколько маркеров. Данный этап представляет наибольшую сложность, поскольку его нельзя автоматизировать.

***Этап 2 – Проверка маркеров***

Проверка маркеров была осуществлена путем выравнивания аминокислотной последовательности белка против референсных геномов представителей микробиоты кишечника.

tblastn -query markername -subject sf/smpdata1/kropachev/data/genome\_base/Refbacterialname.fna

***Этап 3 – Сборка транскриптома***

Сборка транскриптома *de novo* была выполнена с использованием платформы Trinity для работы с данными RNA-Seq в немодельных организмах. Использовались параметры, рекомендуемые по умолчанию для неспаренных прочтений, были использованы все образцы:

Trinity --seqType fq --samples\_file samples.txt --max\_memory 10G --CPU 6

***Этап 4 – Получение данных об обилии транскриптов***

Мы предполагаем, что нуклеотидные последовательности контигов в сборке соответствуют последовательностям транскриптов. Поэтому для получения данных о представленности транскриптов мы использовали программу Kallisto. Для полученной сборки было проведено индексирование последовательностей – проставление определенным участкам генома индексов, использование которых позволяет быстрее выполнять выравнивания за счет сжатия информации. Затем были проведены псевдовыравнивания (оптимизированный алгоритм на основе использования k-меров) для каждого образца на индексированную сборку. Везде использовались параметры по умолчанию. Данные о об обилии транскриптов представлены в виде TPM – значений, выровненных на контиг ридов, нормированных на длину гена и глубину секвенирования.

kallisto index -i Trinity.index /home/kropachev/Work/Trinity\_work/trinity\_out\_dir/Trinity.fasta

kallisto quant -i Trinity.index --single -o abundance/experimentN.Kal -l 52 -s 1 sf/smpdata1/kropachev/data/SRR4841990.fastq

где experiment – название типа диеты, а N – номер элемента выборки для такой диеты.



***Этап 5 – Поиск маркерных последовательностей в сборке транскриптома***

Работа производилась в отдельной директории для каждого маркера.

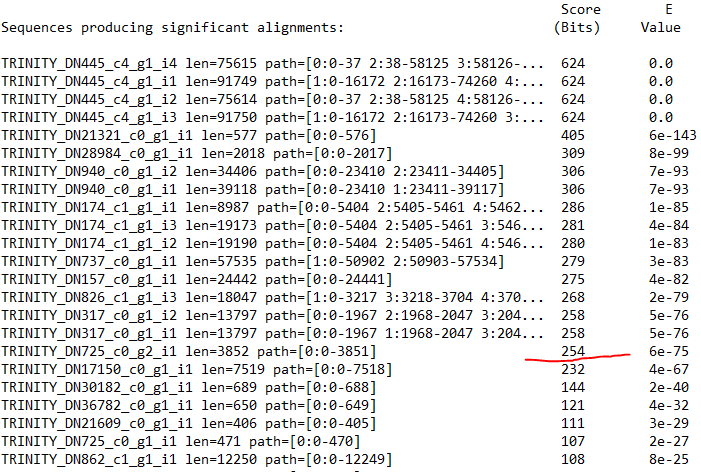
Было проведено выравнивание всех аминокислотных последовательностей ферментов (маркеров) против сборки транскриптома.

tblastn -query markername -subject /home/kropachev//Work/Trinity\_work/trinity\_out\_dir/Trinity.fasta > tblastn\_markername

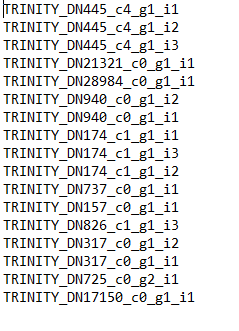
**С этого момента идем в одну папку.**

Все контиги со значениями Score для выравнивания больше 250 были включены в дальнейший анализ. Отсев производился при помощи скрипта findContigID-def.py:

python ../findContigfID-def.py



При помощи скрипта были отобраны ID контигов в файл ContigID\_markername.txt:



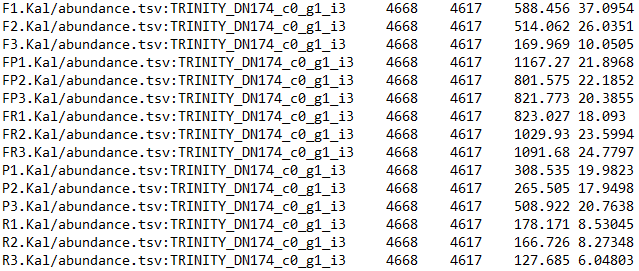
Также был создан файл ContigID\_markername\_dict.txt, где хранится информация о соответствии ID контигов их номерам.

***Этап 6 – Поиск значений TPM для контигов, содержащих маркерный ген***

Значения TPM для контигов, содержащих маркерный ген, были отобраны в данных Kallisto, полученных на предыдущих этапах работы. Для этого для каждого номера контига проводим поиск:

grep -f ContigID\_markername.txt ../abundance/\*.Kal/abundance.tsv > rawTPM\_markername.txt

Получаем файл следующего типа:

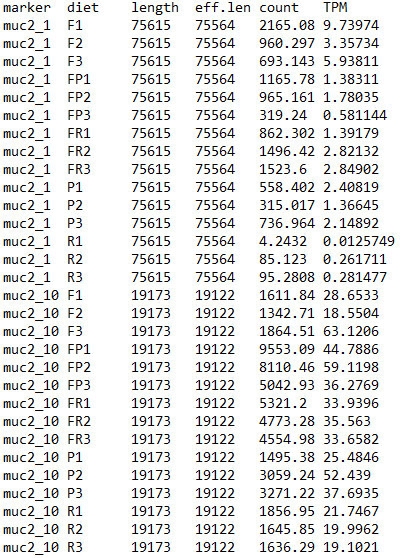


Где последовательность столбцов следующая: target\_id, length, eff\_length, est\_counts, tpm

***Этап 7 – Обработка результатов***

python ../TPMformater-def.py

Объединяем все в один файл таким образом, чтобы все контиги были пронумерованы как nameN\_M, где name – название функциональной группы, N – номер маркера, а M – номер контига:



При помощи скрипта StatUp-def.py значения TPM контигов отдельных маркеров были суммированы для каждого биологического образца (F1, F2, R1…). Далее значения TPM были объединены в выборки по типам диет (3 образца в выборке). Для таких выборок были подсчитаны средние значения TPM, нарисованы графики изменений представленности функциональных групп в зависимости от типов диет.

Также мы сравнили данные об относительном обилии микроорганизмов, построенных на основе 16S рРНК из статьи Десаи с данными о представленности маркеров для каждой из диет. Для того чтобы получить возможность сравнения этих данных, мы суммировали значения обилия для видов, принадлежащих к одной и той же функциональной группе.

На последнем этапе работы мы сравнили динамику изменения представленности функциональных групп от условий среды, определённую нашим методом, с динамикой, которая была получена из данных Десаи. Представленность функциональных групп в нашем методе мы определяли по одному лучшему маркеру для каждой функциональной группы. Мы использовали: acet1, sulfat1, but1, muc2.