Все задания практикумов 11-13 выполнялись в папке /mnt/scratch/NGS/labanovvlad/pr11-13

Все задания практикума 14 выполнялись в папке /mnt/scratch/NGS/labanovvlad/pr14

# Пр. 11

## Подготовка референса

### Получение референса

Референсная хромосома (3) была копирована для дальнейшей работы:

```
mkdir ref
cd ref
cp /mnt/scratch/NGS/DATA/hg38/Homo_sapiens.GRCh38.dna.chromosome.3.fa chr3.fa
```

### Индексация для hisat2

Хромосома была проиндексирована для hisat2 (полученные файлы были помещены в подпапку indexed):

```
hisat2-build chr3.fa indexed/chr3
```

Данная программа принимает имя файла с референсным геномом (chr3.fa) и basename для индекс-файлов и создает бинарники вида basename.N.ht2 (в моем случае indexed/chr3.N.ht2, где N от 1 до 8).

### Индексация samtools

Хромосома была проиндексирована для samtools:

```
samtools faidx chr3.fa
```

Программа принимает имя файла с референсным геномом (chr3.fa) и создает файл (chr3.fa.fai), содержащий строку: “3 198295559 56 60 61”, где 3 – номер хромосомы, 198295559 – ее длина (в нуклеотидах), 56 – номер байта начала последовательности (байты до него - заголовок), 60 – кол-во нуклеотидов в строке, 61 – кол-во байтов в строке (т.е. добавляется ‘\n’).

## Риды ДНК

### Описание образца

a. ID: SRR10720407
b. Ссылка: https://www.ncbi.nlm.nih.gov/sra/SRR10720407
c. Прибор: Illumina Genome Analyzer IIx
d. Организм: Homo sapiens
e. Стратегия: whole-exome sequencing (экзомное)
f. Парно-концевые риды
g. Ожидаемое кол-во ридов (spots): 38530707

### Проверка качества исходных ридов

Риды были копированы для дальнейшей работы:

```
cd ..
mkdir reads
cd reads
cp /mnt/scratch/NGS/DATA/dna_reads/SRR10720407_1.fastq.gz SRR10720407_1f.fq.gz
cp /mnt/scratch/NGS/DATA/dna_reads/SRR10720407_2.fastq.gz SRR10720407_2r.fq.gz
```

Далее были получены файлы для анализа качества ридов:

```
fastqc SRR10720407_1f.fq.gz SRR10720407_2r.fq.gz
```

Программа принимает имена файлов с ридами (`SRR10720407_1f.fq.gz` и `SRR10720407_2r.fq.gz`) вида smth.fq.gz и создает файлы вида smth_fastqc.zip и smth_fastqc.html (`SRR10720407_1f_fastqc.zip`, `SRR10720407_2r_fastqc.zip`, `SRR10720407_1f_fastqc.html`, `SRR10720407_2r_fastqc.html`).

a. Кол-во ридов: 38530707

b. Кол-во прямых и обратных ридов совпадает
c. На рисунках 1, 2 показаны графики качества прямых и обратных ридов соответственно:

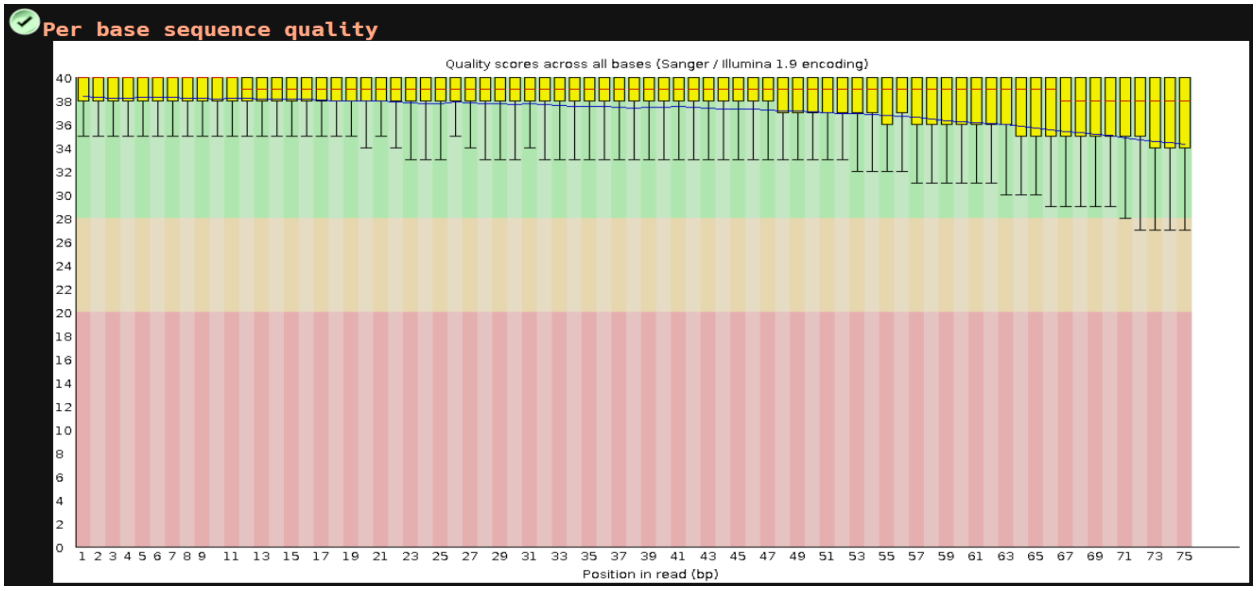

Рис. 1. Качество прямых ридов

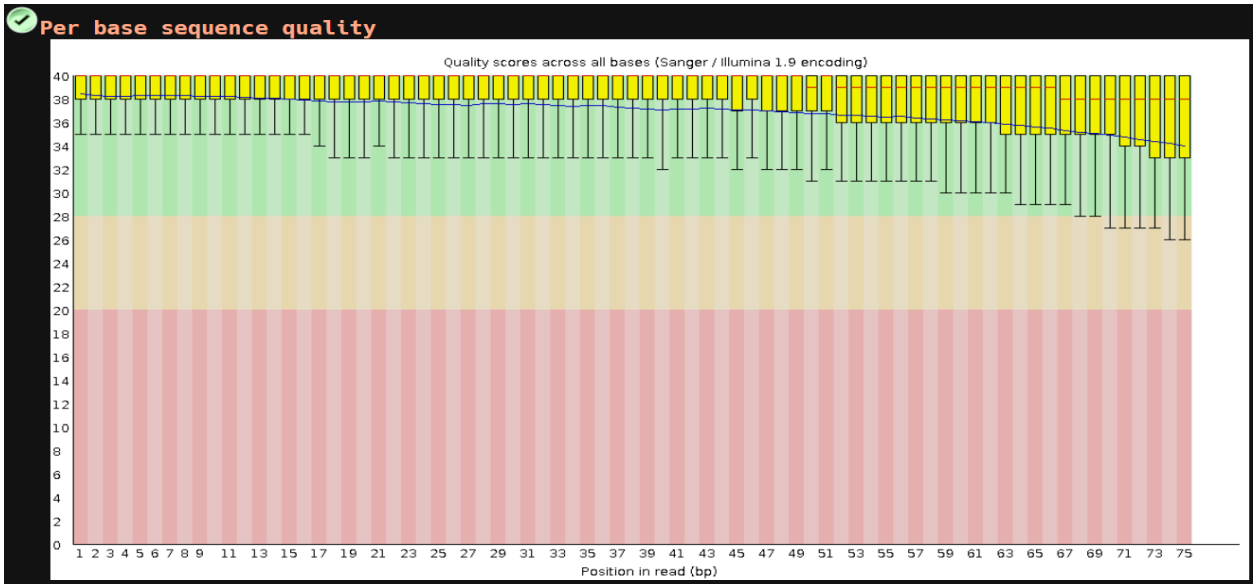

Рис. 2. Качество обратных ридов

Можно заметить, что качество ридов нормальное, но снижается к концу.

d. На рисунках 3, 4 показаны распределения длин прямых и обратных ридов соответственно:

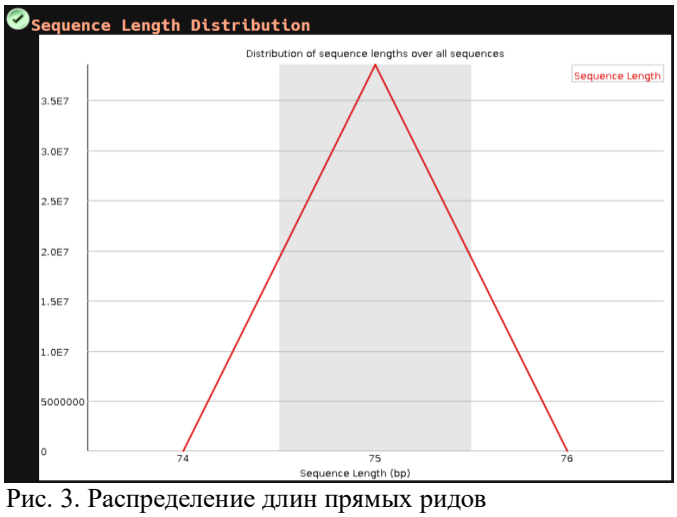

Рис. 3. Распределение длин прямых ридов

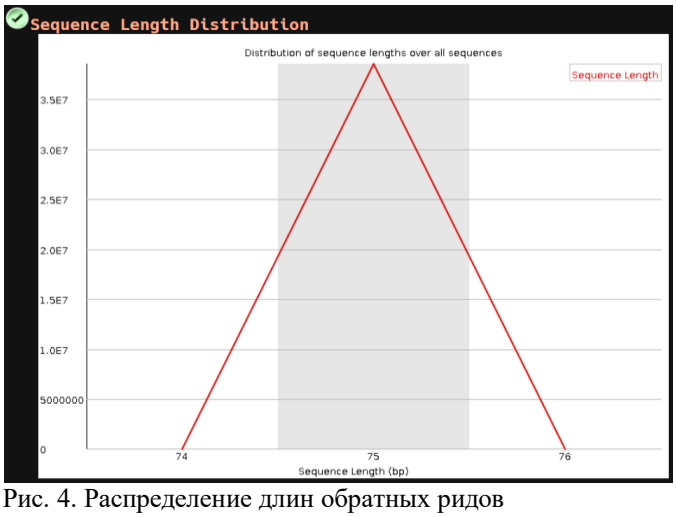

Рис. 4. Распределение длин обратных ридов

Можно заметить, что длина всех ридов 75 нуклеотидов.

## Фильтрация ридов

Чтения были отфильтрованы с помощью программы TrimmomaticPE (т.к. риды парно-концевые):

```
TrimmomaticPE -phred33 -trimlog trimlog.txt SRR10720407_1f.fq.gz SRR10720407_2r.fq.gz
trimmed_1f_paired.fq.gz trimmed_1f_unpaired.fq.gz trimmed_2r_paired.fq.gz
trimmed_2r_unpaired.fq.gz TRAILING:20 MINLEN:50
```

Программа принимает имена файлов с прямыми и обратными ридами (`SRR10720407_1f.fq.gz` и `SRR10720407_2r.fq.gz`) и имена выходных файлов для триммированных парно- и непарно-концевых чтений. Для удаления ридов с качеством ниже 20 был установлен параметр `TRAILING:20`, а для удаления ридов с длиной меньше 50 был установлен параметр `MINLEN:50`. После работы trimmomatic получается 4 файла, т.к. для получения парно-концевых ридов нужно сохранить оба рида в паре, поэтому мы будем использовать только файлы `trimmed_1f_paired.fq.gz` и `trimmed_2r_paired.fq.gz`, а `trimmed_1f_unpaired.fq.gz` и `trimmed_2r_unpaired.fq.gz` могут содержать риды, пара которых была удалена.

## Проверка качества триммированных ридов

Далее были получены файлы для анализа качества триммированных ридов:

```
fastqc trimmed*
```

Принцип работы этой программы уже был описан выше.

a. Кол-во ридов: 37276728
b. Осталось 96.75% ридов
c. На рисунках 5, 6, 7, 8 показаны графики качества пар прямых парных, прямых непарных, обратных парных и обратных непарных ридов соответственно:

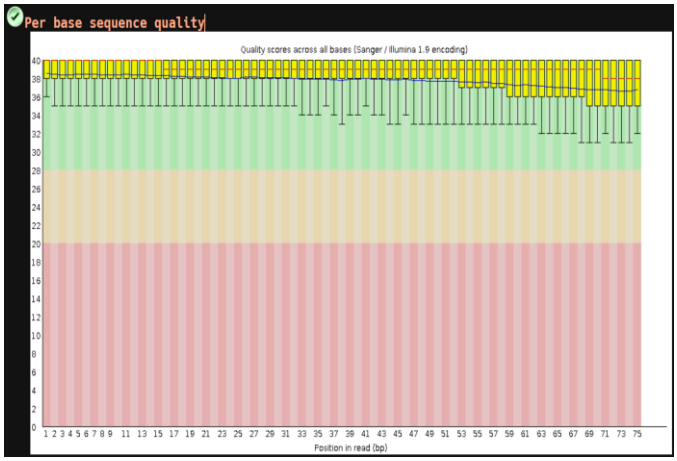

Рис. 5. Качество прямых парных ридов

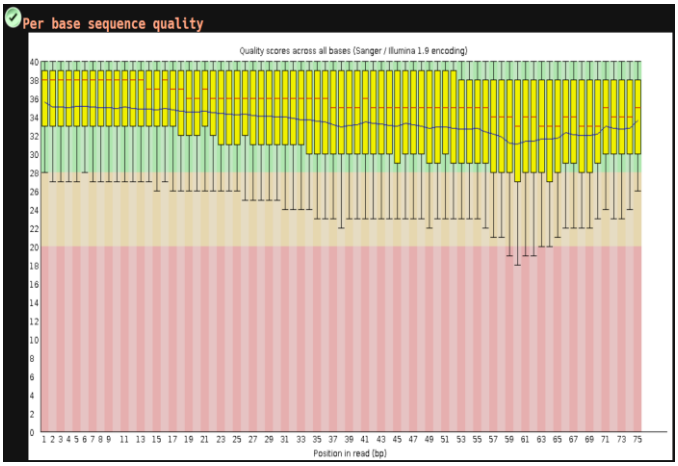

Рис. 6. Качество прямых непарных ридов

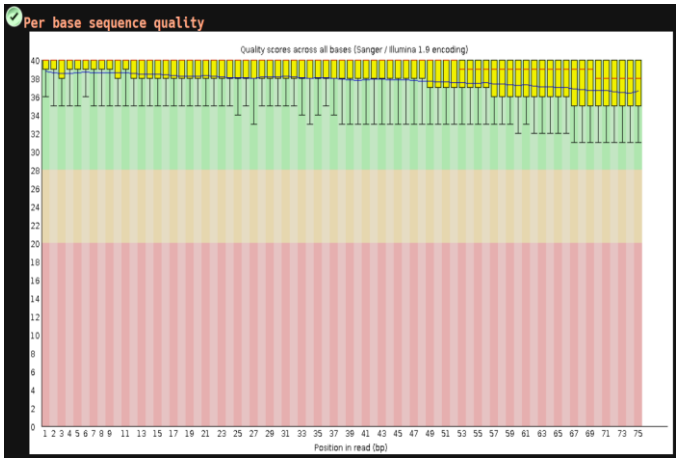

Рис. 7. Качество обратных парных ридов

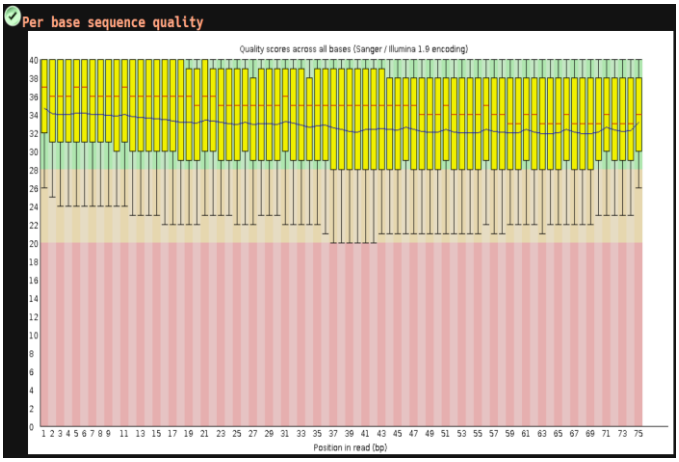

Рис. 8. Качество обратных непарных ридов

Можно заметить, что качество парных ридов высокое (>30), а качество непарных ридов достаточно низкое.

d. Качество ридов значительно повысилось после триммирования.

e. На рисунках 9, 10 показаны распределения длин прямых парных и обратных парных ридов соответственно:

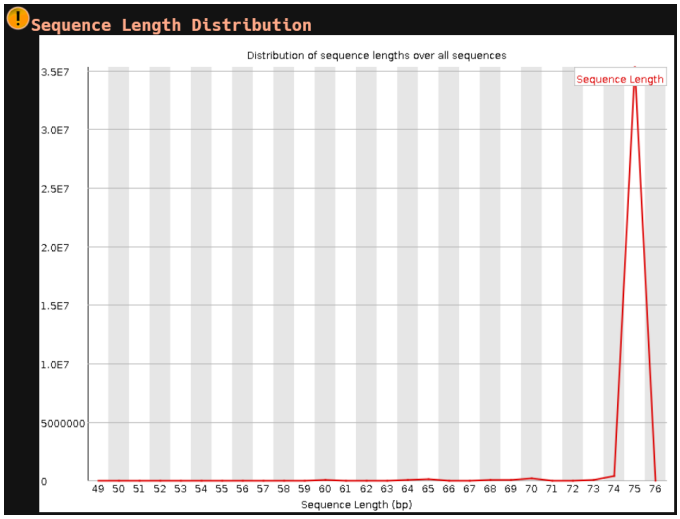


Рис. 9. Распределение длин прямых парных ридов

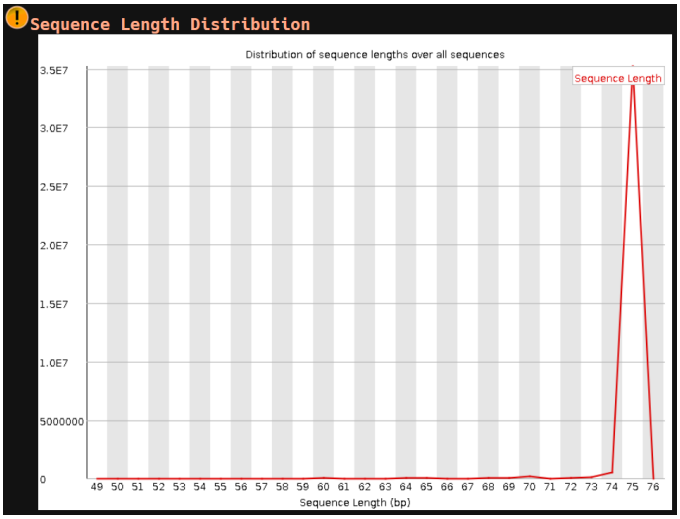


Рис. 10. Распределение длин обратных парных ридов

Можно заметить, что после триммирования появились риды с длиной меньше 75.

# Пр12

## Картирование ридов на референсный геном

Риды были картированы на референсный геном:

```
cd ..
mkdir mapped
cd mapped
hisat2 -x ../ref/indexed/chr3 -1 ../reads/trimmed_1f_paired.fq.gz -2
../reads/trimmed_2r_paired.fq.gz -p 16 --no-spliced-alignment -S paired.sam 2> hisat2_log.txt
```

Программа hisat2 принимает basename индекс-файлов (параметр `-x`), имя файла с прямыми ридами (пар. `-1`), имя файла с обратными ридами (п. `-2`), количество потоков (п. `-p`), имя файла для сохранения результата (п. `-S`). Также был использован параметр `--no-spliced-alignment`, чтобы убрать возможность сплайсинга, а stderr был перенаправлен в лог-файл.

## Конвертация sam в bam

Бинарный bam-файл был получен командой:

```
samtools sort -o paired.bam paired.sam
```

Программа принимает имя sam-файла (`paired.sam`) и создает bam-файл с именем указанным параметром `-o` (`paired.bam`).

a. Файл `paired.sam` весит примерно 15,06 Гб
b. Файл `paired.bam` весит примерно 4,16 Гб

Bam-файл был проиндексирован:

```
samtools index paired.bam
```

Программа принимает имя bam-файла (`paired.bam`) и создает бинарный индекс-файл (`paired.bam.bai`).

## Анализ bam-файла

Был получен файл для анализа:

```
samtools flagstat paired.bam > paired_flagged.txt
```

Программа принимает имя bam-файла (`paired.bam`), output был перенаправлен в `paired_flagged.txt`.

a. Mapped [шт.]: 6297249

b. Mapped [%]: 8.36%
c. Properly paired [шт.]: 4698380
d. Properly paired [%]: 6.30

### Получение ридов, картированных на хромосому 3

Были получены риды, картированные на 3 хромосому:

```
samtools view -h -bS paired.bam 3 > paired_chr3.bam
```

Программа принимает bam-файл (`paired.bam`) и имя хромосомы (3), output был перенаправлен в `paired_chr3.bam`. Также были установлены параметры: включить в файл заголовок (`-h`), output в bam-файл (`-b`), формат input-а определить автоматически (`-S`).

### Получение только правильно картированных пар ридов

Были получены правильно картированные пары ридов:

```
samtools view -f 0x2 -bS paired_chr3.bam > paired_chr3_proper.bam
```

Программа принимает bam-файл (`paired_chr3.bam`), output был перенаправлен в `paired_chr3_proper.bam`. Также были установлены параметры: `-f 0x2`, позволяющий оставить только риды с FLAG = `0x2`, т.е. только PROPER_PAIR; output в bam-файл (`-b`); формат input-а определить автоматически (`-S`).

Далее был получен файл для анализа:

```
samtools flagstat paired_chr3_proper.bam > paired_chr3_proper_flagged.txt
```

Программа принимает имя bam-файла (`paired_chr3_proper.bam`), output был перенаправлен в `paired_chr3_proper_flagged.txt`.

a. Properly paired [шт.]: 4698380
b. Properly paired [%]: 100.00

Bam-файл был проиндексирован:

```
samtools index paired_chr3_proper.bam
```

Программа принимает имя bam-файла и создает бинарный индекс-файл (`paired_chr3_proper.bam.bai`).

# Пр13

### Получение вариантов

Были получены варианты:

```
cd ..
mkdir variants
cd variants
bcftools mpileup -f ../ref/chr3.fa ../mapped/paired_chr3_proper.bam | bcftools call -mv -o
paired_chr3_proper.vcf
```

Программа `mpileup` принимает имя bam-файла (`../mapped/paired_chr3_proper.bam`) и имя файла с референсным геномом (опция `-f`) и создает vcf-файл с вероятностями вариантов, а программа `call` уже собственно ищет варианты и записывает в файл (`paired_chr3_proper.vcf`). Эта программа принимает имя файла для output-а (опция `-o`), использует дефолтный метод поиска (`-m`) и направляет в output только сайты вариантов (`-v`).

Vcf-файл содержит заголовок (каждая строка начинается с “##”) и таблицу с заголовком, который начинается с “#”.

Был получен файл для анализа:

```
bcftools stats paired_chr3_proper.vcf > paired_chr3_proper_stats.txt
```

Программа принимает имя vcf-файла (`paired_chr3_proper.vcf`), output был перенаправлен в `paired_chr3_proper_stats.txt`.

a. Кол-во вариантов: 120061
b. Кол-во SNP: 116331
c. Индели (короткие вставки и замены): 3730

### Фильтрация вариантов

Варианты были отфильтрованы:

```
bcftools filter -i '%QUAL>30 && DP>50' paired_chr3_proper.vcf > paired_chr3_proper_filtered.vcf
```

Программа принимает имя vcf-файла (`paired_chr3_proper.vcf`) и критерии фильтрации (параметр `-i`). Стоит указать, что качество (`QUAL`) является одним из столбцов таблицы vcf-файла, а длина (`DP`) является одним из значений, указанных в столбце `INFO`. Output был перенаправлен в файл `paired_chr3_proper_filtered.vcf`.

Был получен файл для анализа:

```
bcftools stats paired_chr3_proper_filtered.vcf > paired_chr3_proper_filtered_stats.txt
```

Программа принимает имя vcf-файла (`paired_chr3_proper_filtered.vcf`), output был перенаправлен в `paired_chr3_proper_filtered_stats.txt`.

a. Кол-во вариантов: 1918 (1.6%)
b. Кол-во SNP: 1853 (1.59%)
c. Индели (короткие вставки и замены): 65 (1.74%)

### Аннотация вариантов

Файл `paired_chr3_proper_filtered.vcf` был проанализирован с помошью сервиса VEP. Информация из раздела Summary statistics приведена на рисунке 11.

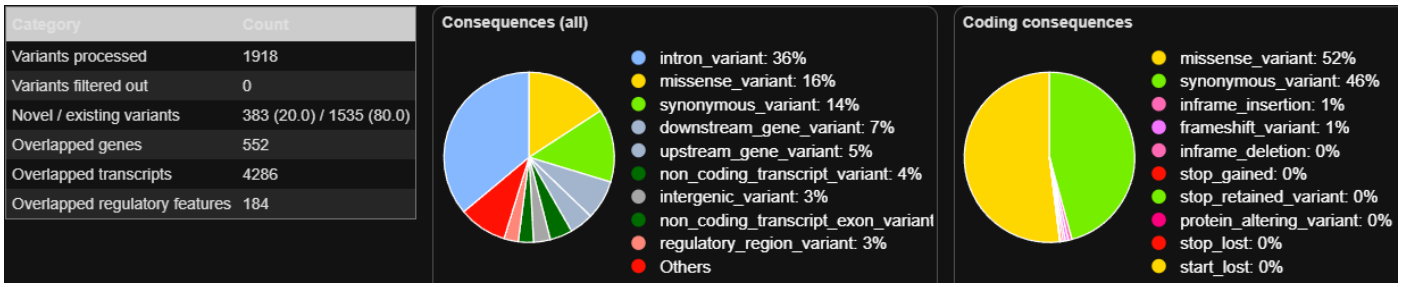

Рис. 11. VEP Summary statistics

- Кол-во вариантов с IMPACT HIGH: 56
- Кол-во вариантов с IMPACT MODIFIER: 8743
- Кол-во вариантов с IMPACT LOW: 2668
- Кол-во вариантов с IMPACT MODERATE: 2501

# Пр14

### Описание образца

a. ID: ENCFF975AUW
b. Ссылка: https://www.encodeproject.org/files/ENCFF975AUW/
c. Организм и ткань: Homo sapiens heart tissue male embryo (120 days)
d. Стратегия: polyA plus RNA-seq
e. Одно-концевые риды
f. Цепь-специфичность: unstranded

### Проверка качества исходных ридов

Риды были копированы для дальнейшей работы:

```
mkdir reads
cd reads
cp /mnt/scratch/NGS/DATA/rna_reads/ENCFF975AUW.fastq.gz ENCFF975AUW.fq.gz
```

Далее были получены файлы для анализа качества ридов:

```
fastqc ENCFF975AUW.fq.gz
```

Программа принимает имя файла с ридами (`ENCFF975AUW.fq.gz`) вида smth.fq.gz и создает файлы вида smth_fastqc.zip и smth_fastqc.html (`ENCFF975AUW_fastqc.zip`, `ENCFF975AUW_fastqc.html`).

a. Кол-во ридов: 87265266
b. На рисунке 12 показан график качества ридов. Можно заметить, что качество ридов очень низкое.
c. На рисунке 13 показано распределение длин ридов. Можно заметить, что длина всех ридов 36 нуклеотидов.

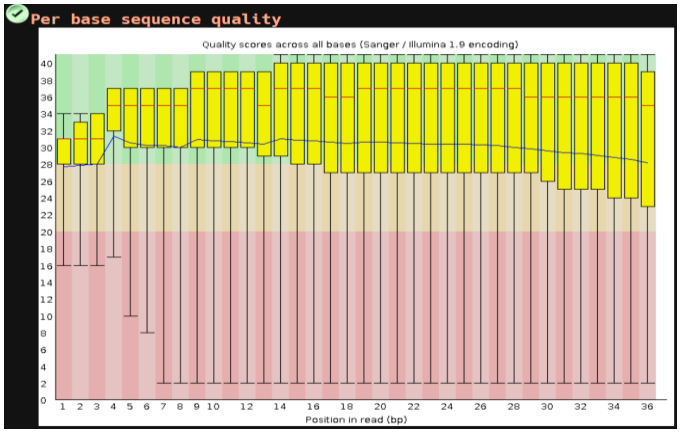

Рис. 12. Качество ридов

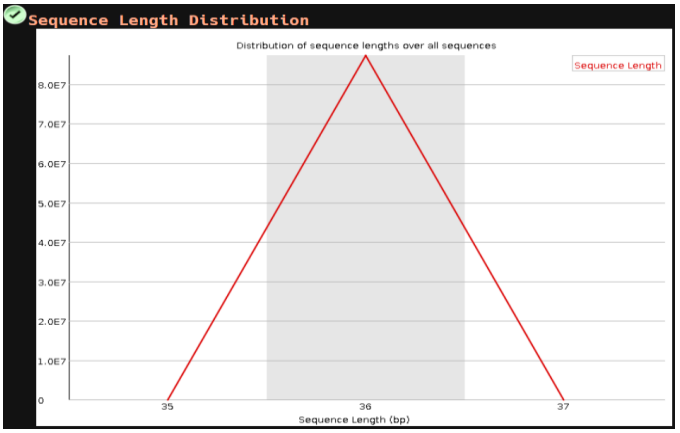

Рис. 13. Распределение длин ридов

## Картирование ридов на референс

Риды были картированы на референсный геном:

```
cd ..
mkdir mapped
cd mapped
hisat2 -x ../../pr11-13/ref/indexed/chr3 -k 3 -U ../reads/ENCFF975AUW.fq.gz -S rna.sam
2>hisat2_log.txt
```

Программа hisat2 принимает basename индекс-файлов (параметр -x), имя файла с ридами (пар. -U), имя файла для сохранения результата (п. -S), также был использован параметр –k 3, который означает, что программа будет искать по 3 выравнивания для каждого рида (причем таких выравнивания, что score каждого >= score любого другого выравнивания). Stderr был перенаправлен в лог-файл.

a. Кол-во закартировавшихся ридов: 6689124 (7.67%)

Бинарный bam-файл был получен командой:

```
samtools sort -o rna.bam rna.sam
```

Bam-файл был проиндексирован:

```
samtools index rna.bam
```

Были получены риды, картированные на 3 хромосому:

```
samtools view -h -bS rna.bam 3 > rna_chr3.bam
```

Bam-файл был проиндексирован:

```
samtools index rna_chr3.bam
```

Далее был получен файл для анализа:

```
samtools flagstat rna_chr3.bam > rna_chr3_flagged.txt
```

## Поиск экспрессирующихся генов

Файл с генной разметкой был копирован для дальнейшей работы:

```
cd ..
```

```
mkdir marking
cp /mnt/scratch/NGS/DATA/genes/Homo_sapiens.GRCh38.110.chr.gtf marking/marking.gtf
cd mapped
```

Gtf-файл содержит заголовок (каждая строка начинается с “#”) и таблицу особенностей.

Для каждого гена из разметки было посчитано кол-во картированных на этот ген ридов:

```
htseq-count -f bam -s no -m union -t exon -o marked_rna_chr3.sam rna_chr3.bam
../marking/marking.gtf 1> marked_rna_chr3.txt 2> htseq_count_log.txt
```

Программа htseq-count принимает имена bam-файла, файла с генной разметкой и файла с результатом подсчета (пар. `-o`), расширение входного файла (п. `-f`) и тип гена из разметки (п. `-t`), только гены нужного типа (т.е. `exon`) будут анализироваться. Также был использован параметр `-s no`, который означает, что программа будет считать риды, попадающие и на прямую, и на обратную цепь и параметр `-m union`, который означает, что программа будет объединять перекрывающиеся риды. Stdout и stderr были перенаправлены в лог-файл.

Из лог-файла мы можем узнать, что в нужные гены не попало 1457316 ридов (__no_feature), в несколько генов попало 231813 ридов (__ambiguous), также обнаружилось 753766 ридов, для которых трудно однозначно определить соответствующий ген (__alignment_not_unique). В получившемся после работы программы sam-файле находится таблица, из которой можно узнать, какие риды попали в гены.

Чтобы узнать, сколько ридов попало в границы генов была использована команда:

```
head -n -5 marked_rna_chr3.txt | awk '{sum+=$2} END{print "sum=", sum}'
```

В границы генов попало 4246229 ридов.

# Подготовка программного сценария

Программный сценарий расположен в файле `/mnt/scratch/NGS/labanovvlad/pr11-13/NGS_script.sh`.

NGS_script.sh:

```
#!/bin/bash

# Usage: ./NGS_script.sh ID N

#############################################
#  Soft
#############################################
#  HISAT2 version 2.2.1
#  samtools 1.17 (using htslib 1.17)
#  FastQC v0.11.9
#  TrimmomaticPE 0.39
#  bcftools 1.11 (using htslib 1.11-4)
#############################################

if [[ "$1" == "-h" ]] || [[ "$1" == "--help" ]]; then
        echo "Usage: ./NGS_script.sh ID N" && exit 0
fi

ID=$1
N=$2
#config=${3:-"./config.txt"}

#. $config

[ -d ${2}_$1 ] && echo "please, rename directory ${2}_$1 (I wanna put files on that address)" &&
exit 1

mkdir ${2}_$1
cd ${2}_$1
mkdir ref
cd ref
cp /mnt/scratch/NGS/DATA/hg38/Homo_sapiens.GRCh38.dna.chromosome.${N}.fa chr${N}.fa
mkdir indexed
hisat2-build chr${N}.fa indexed/chr$N
```

```
samtools faidx chr${N}.fa
cd ..
mkdir reads
cd reads
cp /mnt/scratch/NGS/DATA/dna_reads/${ID}_1.fastq.gz ${ID}_1f.fq.gz
cp /mnt/scratch/NGS/DATA/dna_reads/${ID}_2.fastq.gz ${ID}_2r.fq.gz
fastqc ${ID}_1f.fq.gz ${ID}_2r.fq.gz
TrimmomaticPE -threads 16 -phred33 -trimlog trimlog.txt ${ID}_1f.fq.gz ${ID}_2r.fq.gz
trimmed_1f_paired.fq.gz trimmed_1f_unpaired.fq.gz trimmed_2r_paired.fq.gz
trimmed_2r_unpaired.fq.gz TRAILING:20 MINLEN:50
fastqc trimmed*
cd ..
mkdir mapped
cd mapped
hisat2 -x ../ref/indexed/chr$N -1 ../reads/trimmed_1f_paired.fq.gz -2
../reads/trimmed_2r_paired.fq.gz -p 16 --no-spliced-alignment -S paired.sam 2> hisat2_log.txt
samtools sort -o paired.bam paired.sam
#rm paired.sam
samtools index paired.bam
samtools flagstat paired.bam > paired_flagged.txt
samtools view -h -bS paired.bam $N > paired_chr${N}.bam
samtools view -f 0x2 -bS paired_chr${N}.bam > paired_chr${N}_proper.bam
samtools flagstat paired_chr${N}_proper.bam > paired_chr${N}_proper_flagged.txt
samtools index paired_chr${N}_proper.bam
cd ..
mkdir variants
cd variants
bcftools mpileup -f ../ref/chr${N}.fa ../mapped/paired_chr${N}_proper.bam | bcftools call -mv -
o paired_chr${N}_proper.vcf
bcftools stats paired_chr${N}_proper.vcf > paired_chr${N}_proper_stats.txt
bcftools filter -i '%QUAL>30 && DP>50' paired_chr${N}_proper.vcf >
paired_chr${N}_proper_filtered.vcf
bcftools stats paired_chr${N}_proper_filtered.vcf > paired_chr${N}_proper_filtered_stats.txt

echo "Done! Your vcf-file is ./${2}_${1}/variants/paired_chr${N}_proper_filtered.vcf"
less ./paired_chr${N}_proper_filtered.vcf
```