Практикумы 11–14: NGS

Практикум 11 3
Подготовка референса 3
1. Индексация для hisat2 3
2. Индексация samtools 3
Подготовка чтений 4
1. Информация об образце 4
2. Проверка качества чтений 4
3. Фильтрация чтений 7
4. Проверка качества триммированных чтений 8
Отчёт о качестве чтений 11
Практикум 12 13
Картирование чтений на референсный геном 13
Конвертация sam файла в bam файл 13
Анализ bam файла 14
Получение чтений, картированных на 12 хромосому 14
Получение только правильно картированных пар чтений 15
Практикум 13 17
Получение вариантов 17
Фильтрация вариантов 18
Аннотация вариантов 19
Практикум 14 20
Описание образца 20
Проверка качества исходных чтений 20
Картирование чтений на референс 23
Поиск экспрессирующихся генов 24

Практикум 11

Подготовка референса

В качестве референса используется последовательность 12 хромосомы из сборки hg38.

1. Индексация для hisat2

Для дальнейшего картирования данных при помощи программы hisat2 нужно сначала индексировать их определённым образом.

```
hisat2-build chr12.fa chr12
```

`hisat2-build` создаёт 8 файлов с указанным префиксом (в данном случае `chr12`) из файла `chr12.fa`. Для геномов короче 4 миллиардов пн (как в нашем случае) он использует 32-битные числа, а для более длинных геномов – 64-битные.

Input: chr12.fa
Output: chr12.1.ht2, …, chr12.8.ht2

2. Индексация samtools

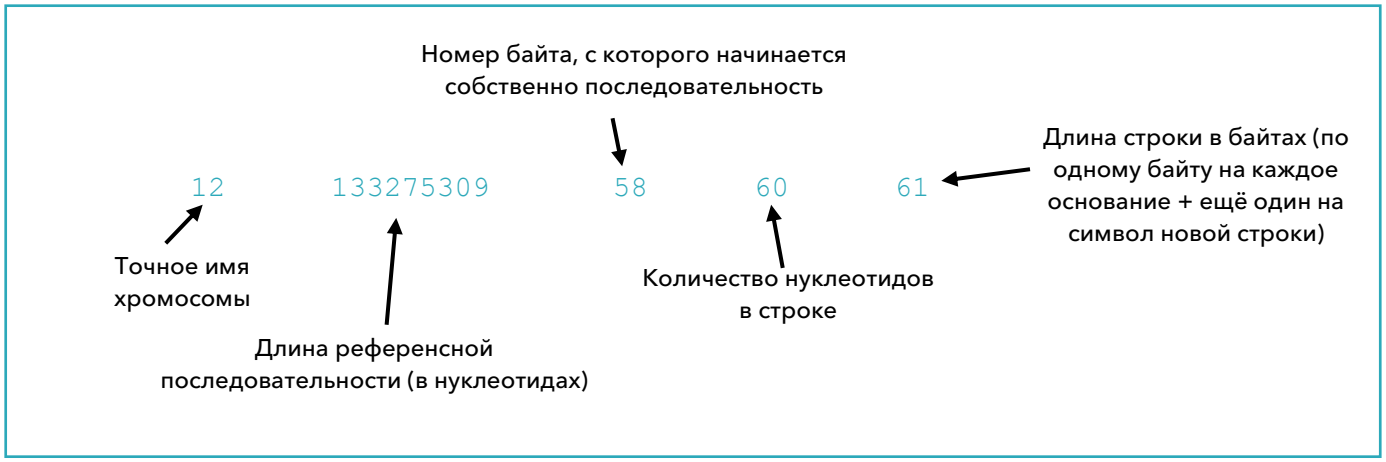
samtools и ещё некоторые программы требуют другой индексации:

```
samtools faidx chr12.fa
```

Насколько я поняла, такой способ создания индекса позволяет проще переходить к нужному месту в референсном геноме без необходимости читать его последовательно, строка за строкой.

Input: chr12.fa
Output: chr12.fa.fai

Содержимое файла chr12.fa.fai выглядит так:



Подготовка чтений

1. Информация об образце

a. SRR ID образца: **SRR10720410**
b. ссылка на информацию об образце из NCBI: воть
c. прибор для секвенирования: **Illumina Genome Analyzer IIx**
d. организм: *Homo sapiens*
e. стратегия секвенирования: в поле strategy указано OTHER, что бы под эти ни подразумевалось, но в более подробном описании протокола сказано про **полноэкзомное секвенирование**
f. вид чтений: **парноконцевые**
g. сколько чтений ожидается (spots): **39,100,758**

2. Проверка качества чтений

FastQC – инструмент для контроля качества данных секвенирования второго поколения.

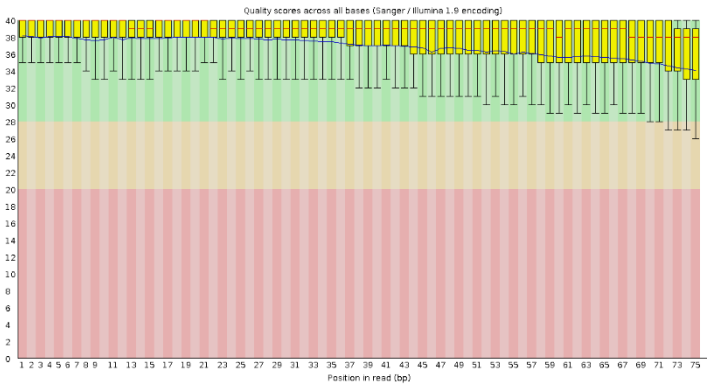
```
fastqc SRR10720410_1.fastq.gz SRR10720410_2.fastq.gz
```

Input: два архива (SRR10720410_1.fastq.gz – с прямыми и SRR10720410_2.fastq.gz – с обратными чтениями)
Output: 2 html файла (ссылки на них будут на странице практикума)

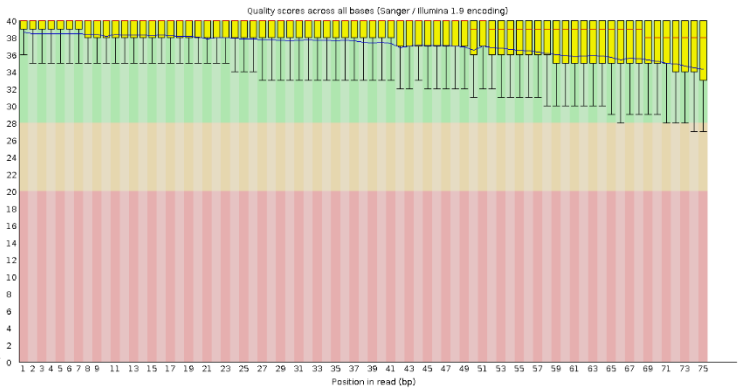
a. Получилось по **39,100,758** чтений в обоих файлах
b. То есть количество прямых и обратных чтений совпадает, и оно также совпадает с ожидаемым количеством
c. **Per base sequence quality:**

Рис. 1 Per base sequence quality **a)** для прямых чтений и **b)** для обратных; медиана обозначена красным, а среднее – синим. Видно, что качество не идеальное, но довольно неплохое (все значительно больше 20), причём качество чтений равномерно снижается к концу.

a)

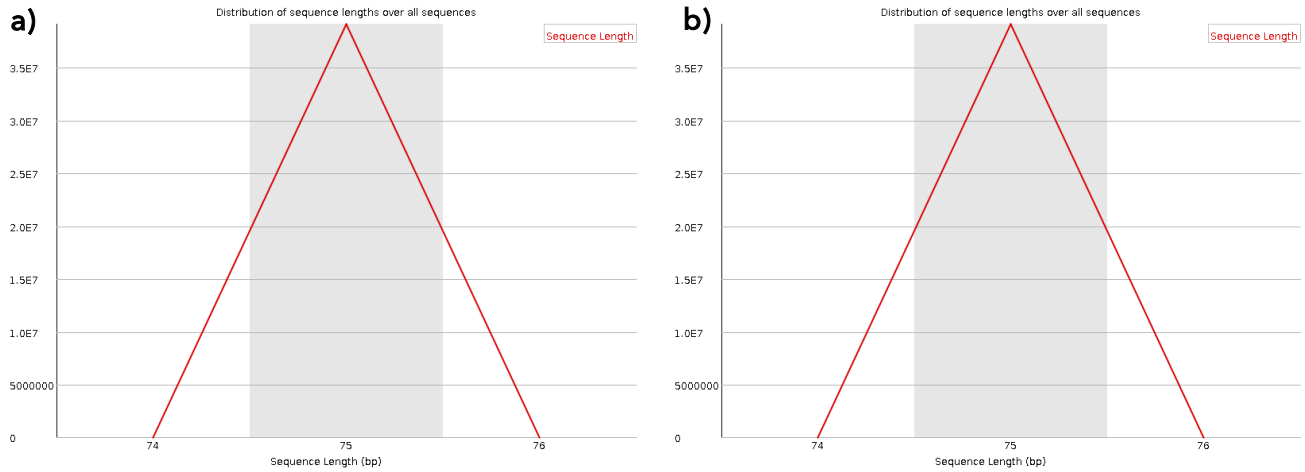


b)



d. Длина чтений равна 75 и для прямых, и для обратных чтений.

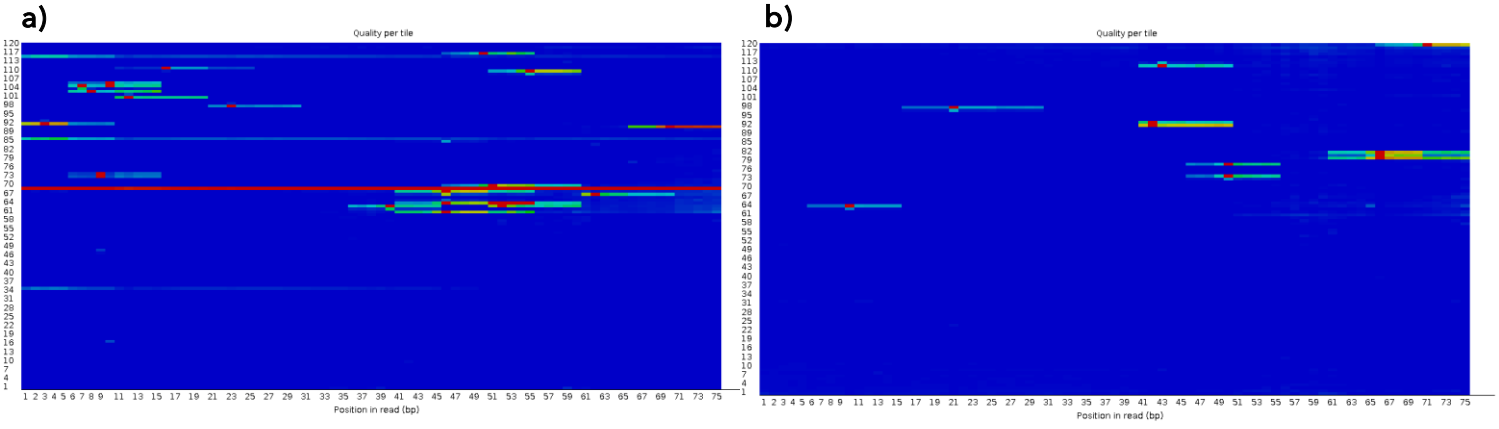
Рис. 2 Распределение чтений по длине **a)** для прямых чтений и **b)** для обратных. Видно, что все чтения имеют длину 75 нуклеотидов (что можно увидеть и из общей статистики тоже, и что типично для данных, полученных с помощью Illumina)



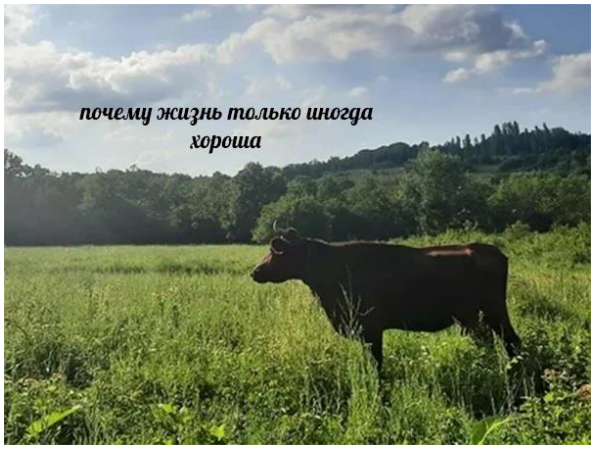
e. Некоторые другие результаты анализа при помощи FastQC

Единственный раздел, в котором программа начала «ругаться» - **per tile sequence quality**. По вертикали в нём отложены номера ячеек, а по горизонтали – позиции в чтениях. Если график имеет синий цвет, то этот участок просеквенировался так же или лучше, чем другие, а вот если он имеет какой-то более тёплый цвет – всё плохо.

Рис. 3 Per tile sequence quality **a)** для прямых чтений и **b)** для обратных. Видно, что при секвенировании прямых чтений одна ячейка вообще целиком выпала



Но это всё ещё не самая ужасная ситуация, так что, наверное, можно не обращать внимания.



Остальные показатели не вызвали у автоматической проверки каких-то вопросов и выглядят примерно одинаково для прямых и обратных чтений, поэму дальше некоторые их них приведены только для прямых:

Рис. 4 Per sequence quality scores (прямые чтения). Для каждого чтения высчитывается среднее значение Q. Виден только один отчётливый пик, что значит, что всё хорошо.

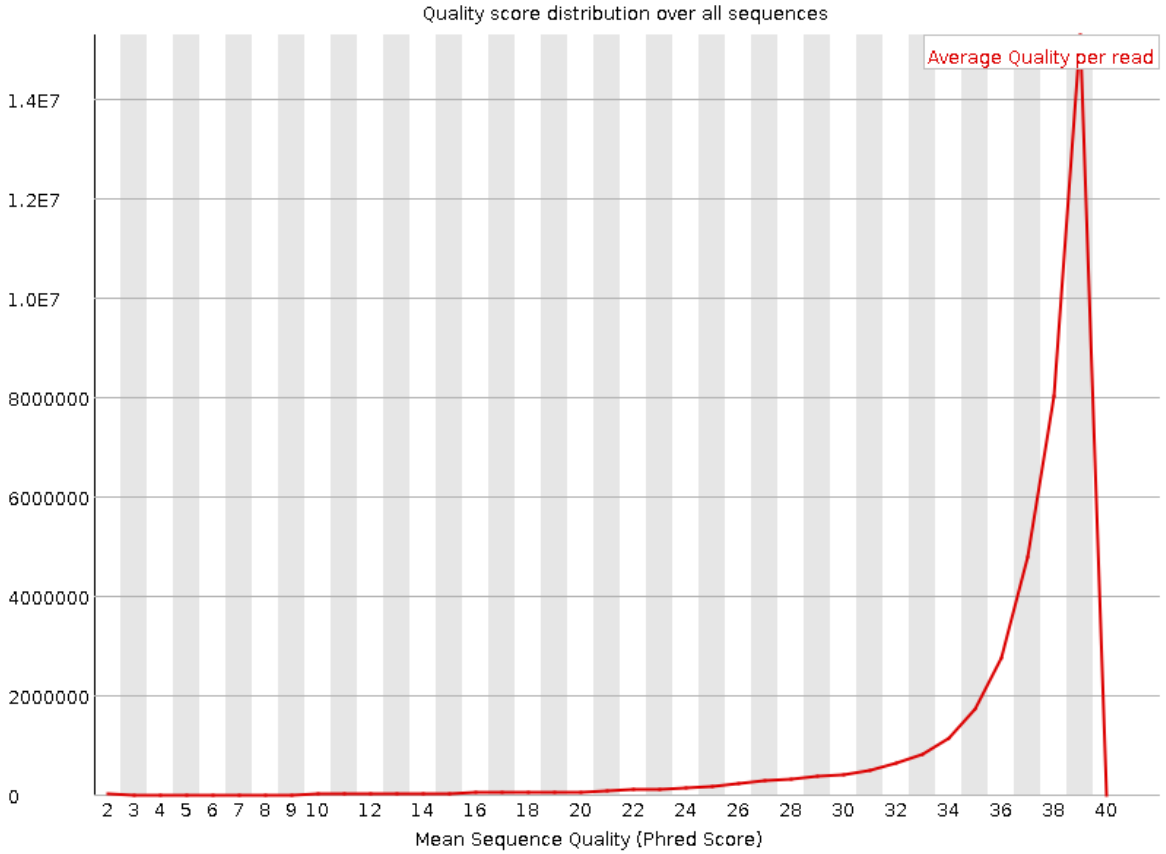


Рис. 5 Per base sequence content (прямые чтения); все линии прямые, как и должно быть, чуть более волнистые только в самом начале.

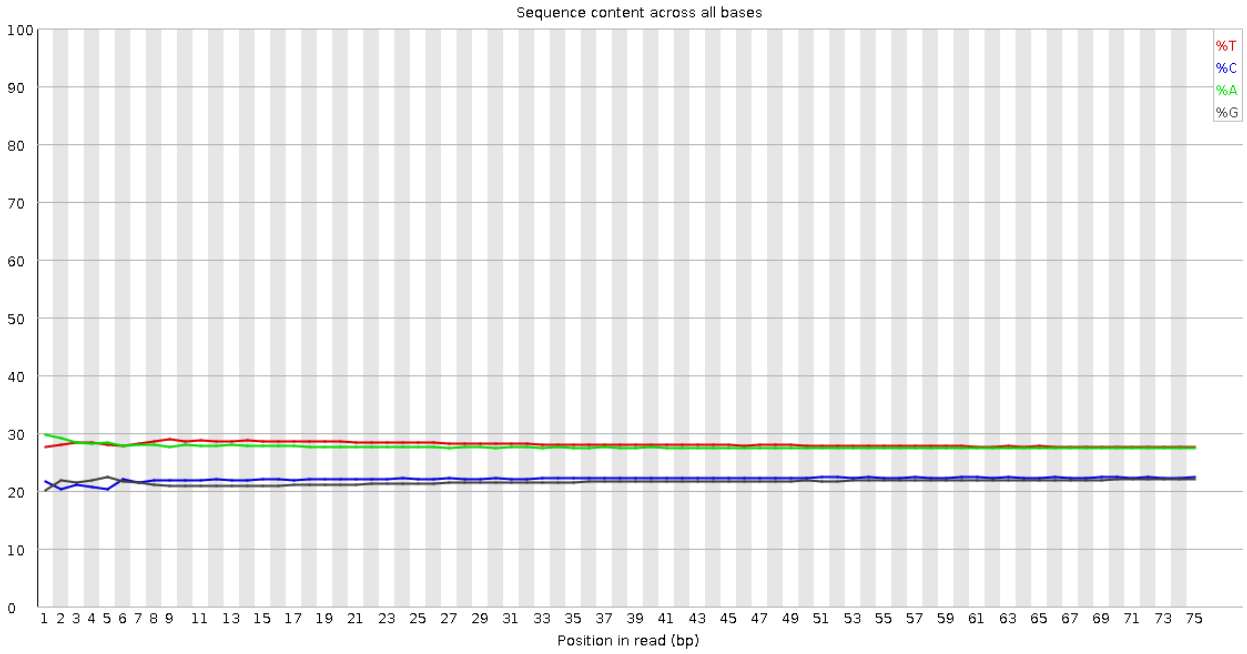
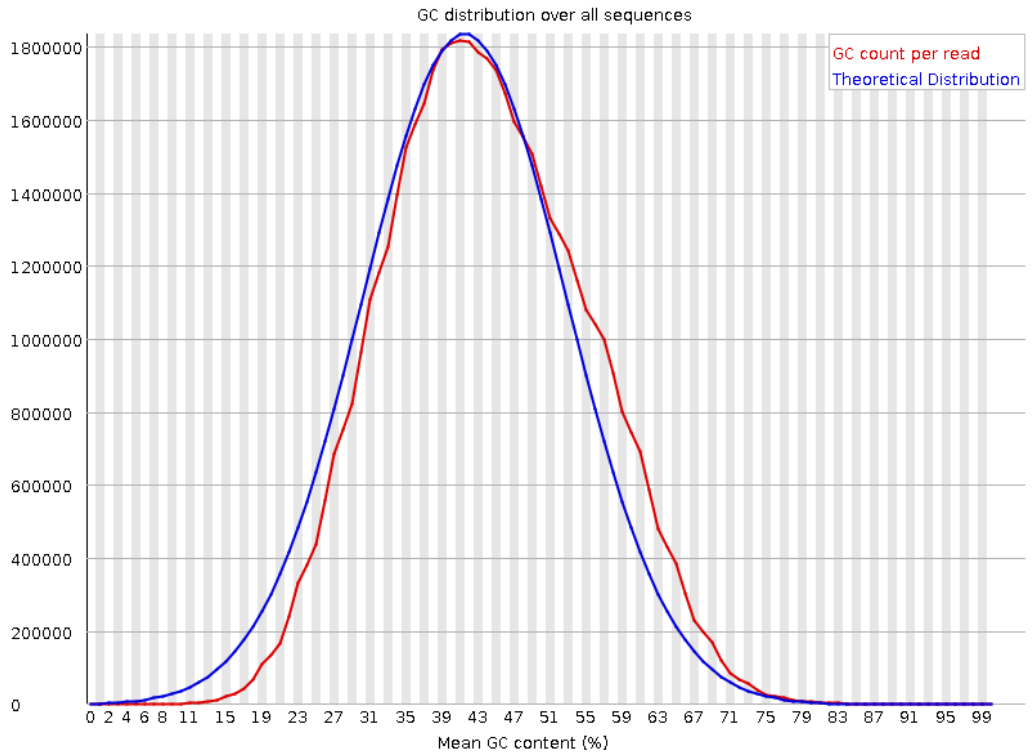


Рис. 6 Per sequence GC content (прямые чтения); синим нарисована ожидаемая гауссова кривая, красным – реальные данные. Можно заметить, что последовательности с различным GC-составом распределены практически идеально нормально.



Также в данных нет перепредставленных последовательностей (тех, которые занимают более 0.1% от всех данных). Если бы они были, можно было бы заподозрить контаминацию.

3. Фильтрация чтений

Вроде как мы обсуждали, что при секвенировании РНК триммировать не очень принято, а тут как раз полноэкзомное секвенирование, но почему бы и нет, впрочем.

```
TrimmomaticPE -phred33 SRR10720410_1.fastq.gz
SRR10720410_2.fastq.gz output_forward_paired.fastq.gz
output_forward_unpaired.fastq.gz
output_reverse_paired.fastq.gz
output_reverse_unpaired.fastq.gz TRAILING:20 MINLEN: 50
```

Важно, чтобы `TRAILING` (удаление нуклеотидов с конца, если их качество хуже порогового) и `MINLEN` (отбрасывание прочтений, если их длина стала меньше пороговой) шли именно в таком порядке, потому что исполнение происходит слева направо.

Input: SRR10720410_1.fastq.gz, SRR10720410_2.fastq.gz
Output: output_forward_paired.fastq.gz, output_forward_unpaired.fastq.gz, output_reverse_paired.fastq.gz, output_reverse_unpaired.fastq.gz

Для парноконцевых чтений Trimmomatic выдает 4 файла: для прямых и для обратных по отдельности есть файлы с теми прочтениями, у которых пара пережила тримминг (paired) и теми, у которых не пережила (unpaired).

4. Проверка качества триммированных чтений

Анализируем результаты тримирования во всех четырёх файлах:

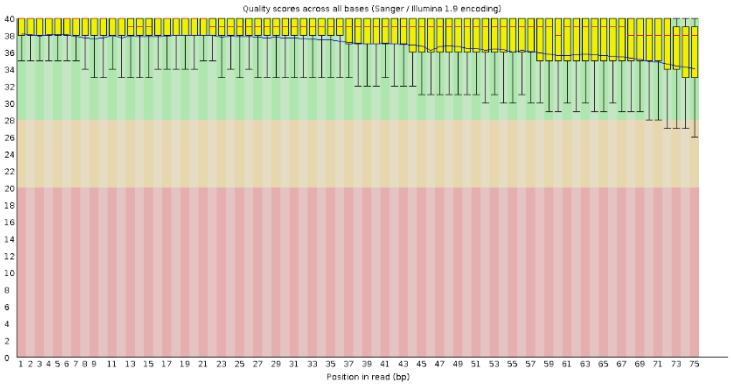
```
fasqc output*.fastq.gz
```

Результаты точно так же будут лежать на странице практикума.

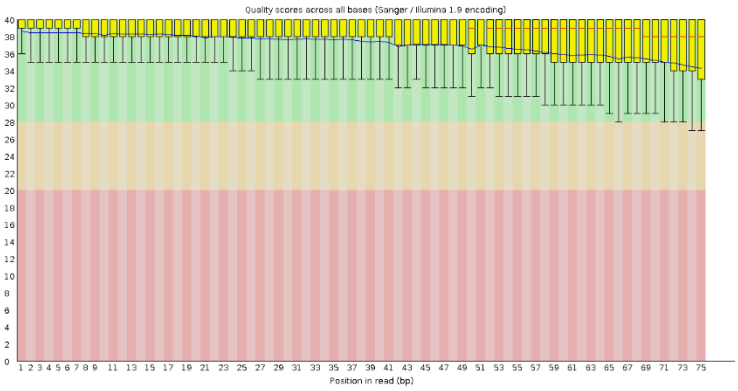
a. Осталось пар: **37,852,184**
b. Процент от исходного количества пар чтений: **96,8%**
c. Если мы сравним качество чтений в парных и непарных чтениях после, увидим следующее:

Рис. 7 Сравнение качества чтений после триммирования

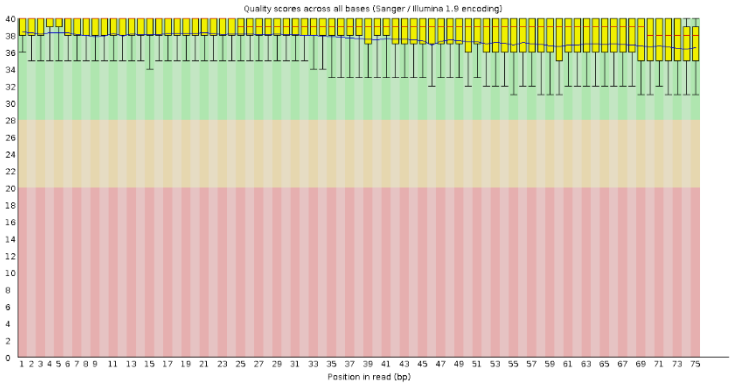
Прямые до



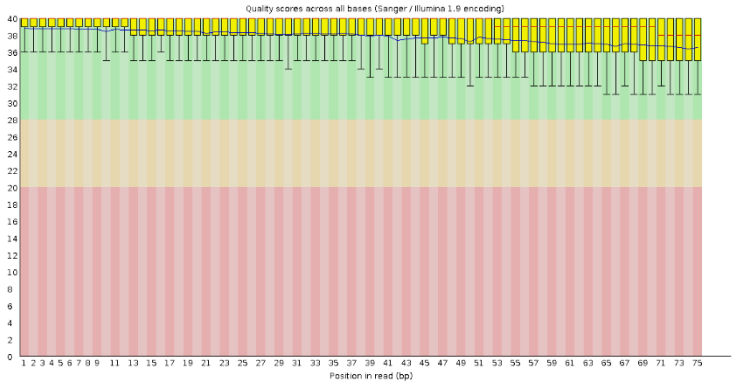
Обратные до



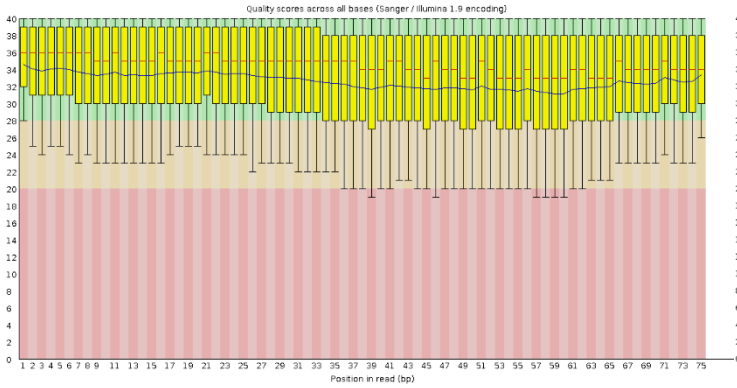
Прямые парные после



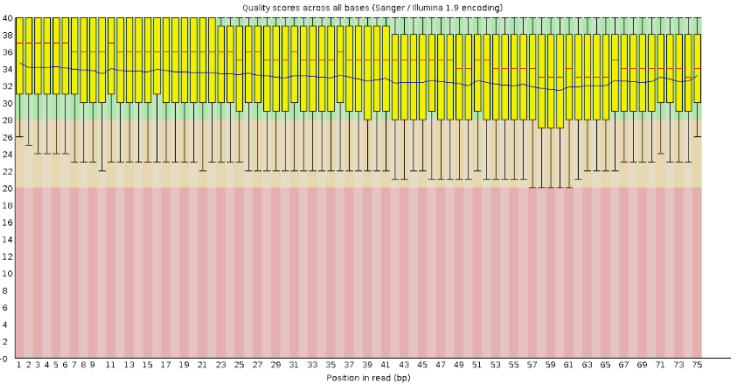
Обратные парные после



Прямые непарные после



Обратные непарные после



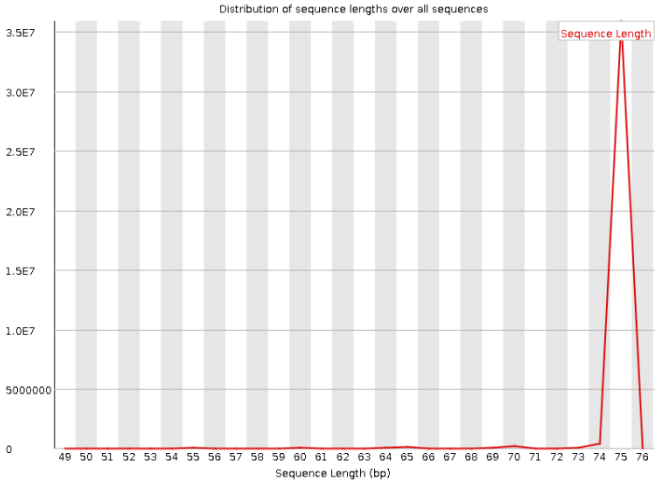
Непарные чтения в полученных файлах оказались сильно хуже по качеству, видимо потому, что изначально были чтения похуже, но один нуклеотид из пары был чуть получше и потому пережил триммирование, хотя и с трудом.

d. Качество чтений среди paired заметно улучшилось, они «прижались» к верхней части графика в самом конце.

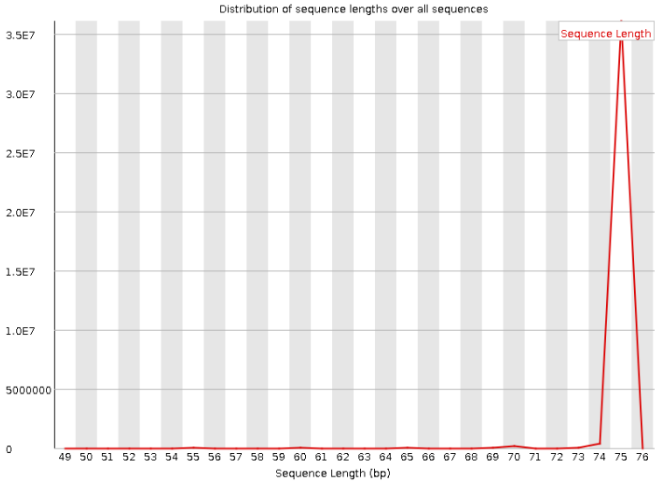
e. Длина чтений стала не одинаковой, и теперь она 50–75 нуклеотидов. Среди парных нуклеотидов длина практически всех так и осталась в основном 75 нуклеотидов, а вот непарные порезались сильнее:

Рис. 8 Изменение длины чтений в результате триммирования

Прямые парные чтения

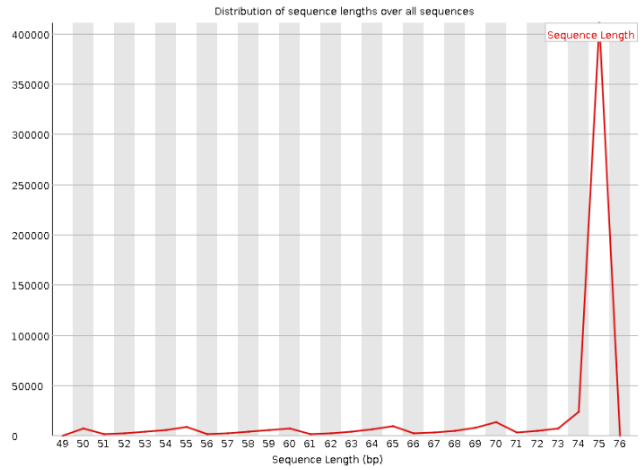


Обратные парные чтения



Прямые непарные чтения

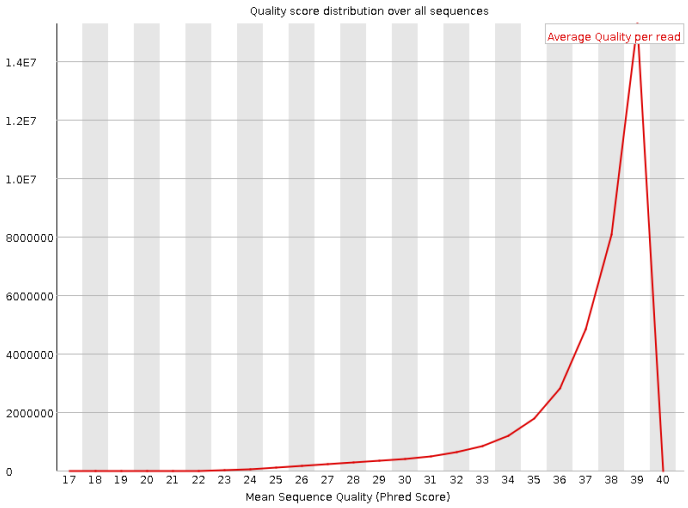
Обратные непарные чтения



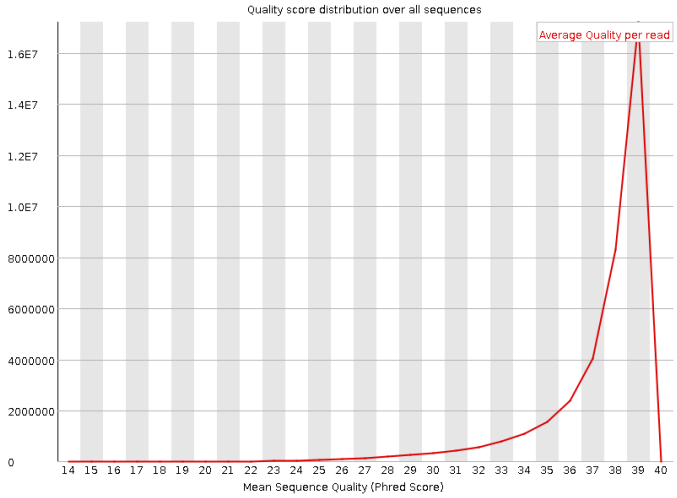
Что примечательно, можно увидеть, что на графике с распределением усреднённых качеств нуклеотидов в чтении в случае непарных чтений появляется второй «горб», и по-хорошему надо выкинуть все чтения со средним качеством ниже, скажем, 35, чтобы его убрать. Либо можно просто выбросить все непарные чтения, их не так уж много.

Рис. 9 Сравнение графиков per sequence quality scores в файлах, полученных в результате триммирования

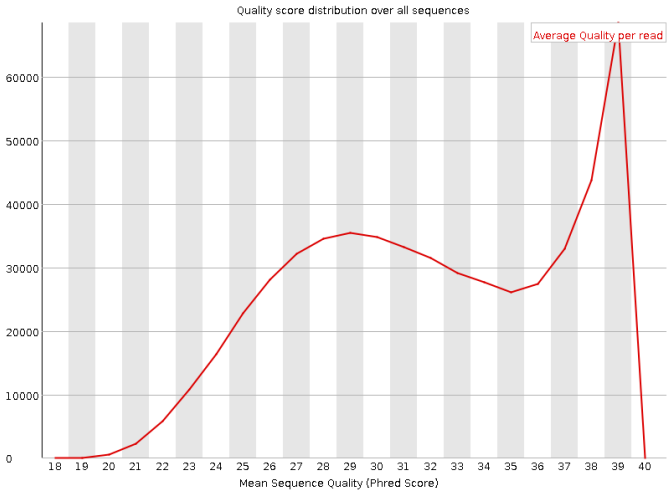
Прямые парные чтения



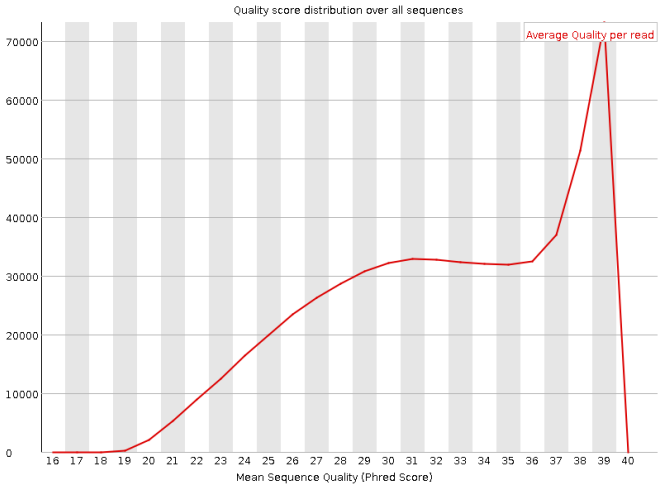
Обратные парные чтения



Прямые непарные чтения



Обратные непарные чтения



Отчёт о качестве чтений

Чтобы как-то упростить себе жизнь, я воспользовалась MultiQC. Насколько я поняла, эта программа собирает информацию из log-файлов других программ и собирает их все в один html файл.

```
mutiqc .
```

Рис. 10 Heatmap, отражающая проверку статуса в разных файлах fastqc; зелёные клеточки – всё хорошо, желтые – есть что-то необычное, красные – есть что-то вообще из ряда вон выходящее. Можно заметить, что per tile sequence quality не починилось в результате триммирования

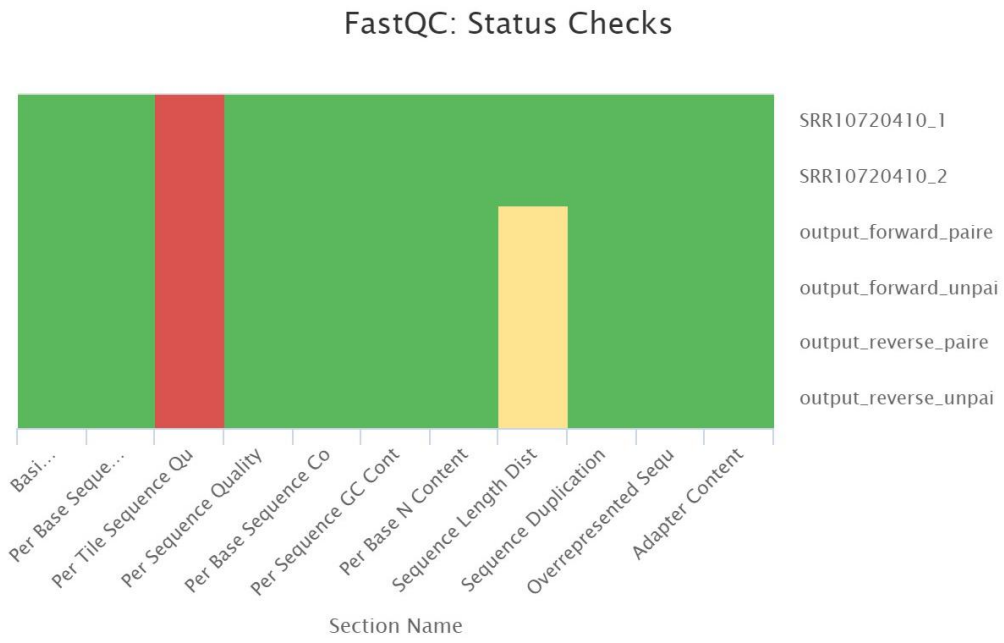


Рис. 11 Per sequence quality scores для всех 6 файлов. Из-за того, что по оси у отложены не доли, а абсолютные количества, качества для неспаренных прочтений приплющились в самом низу, и понятно, что можно, в целом, их просто все отбросить.

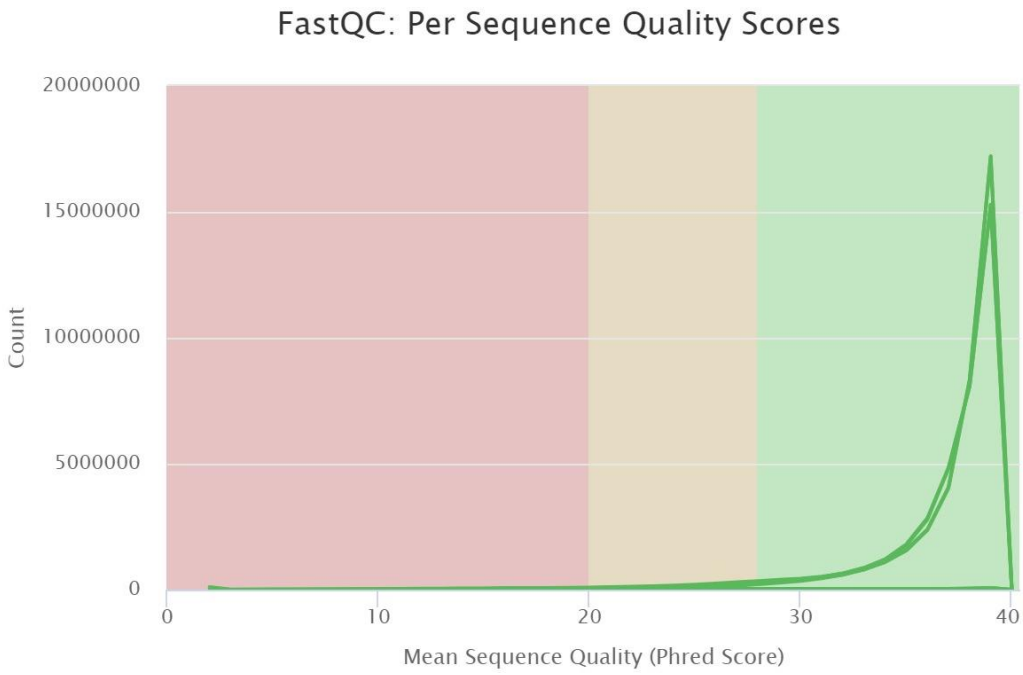


Рис. 12 Количества последовательностей в каждом файле. Здесь также можно визуально оценить, насколько немного (в штуках) последовательностей отбросилось при триммировании, и насколько немного неспаренных проследовательностей при этом получилось

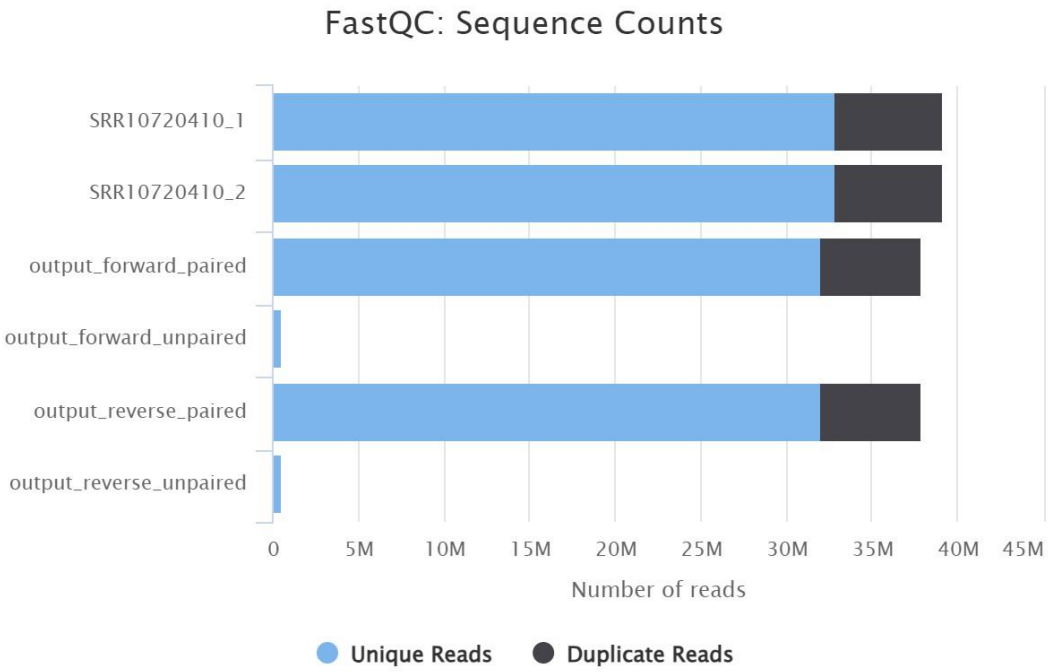
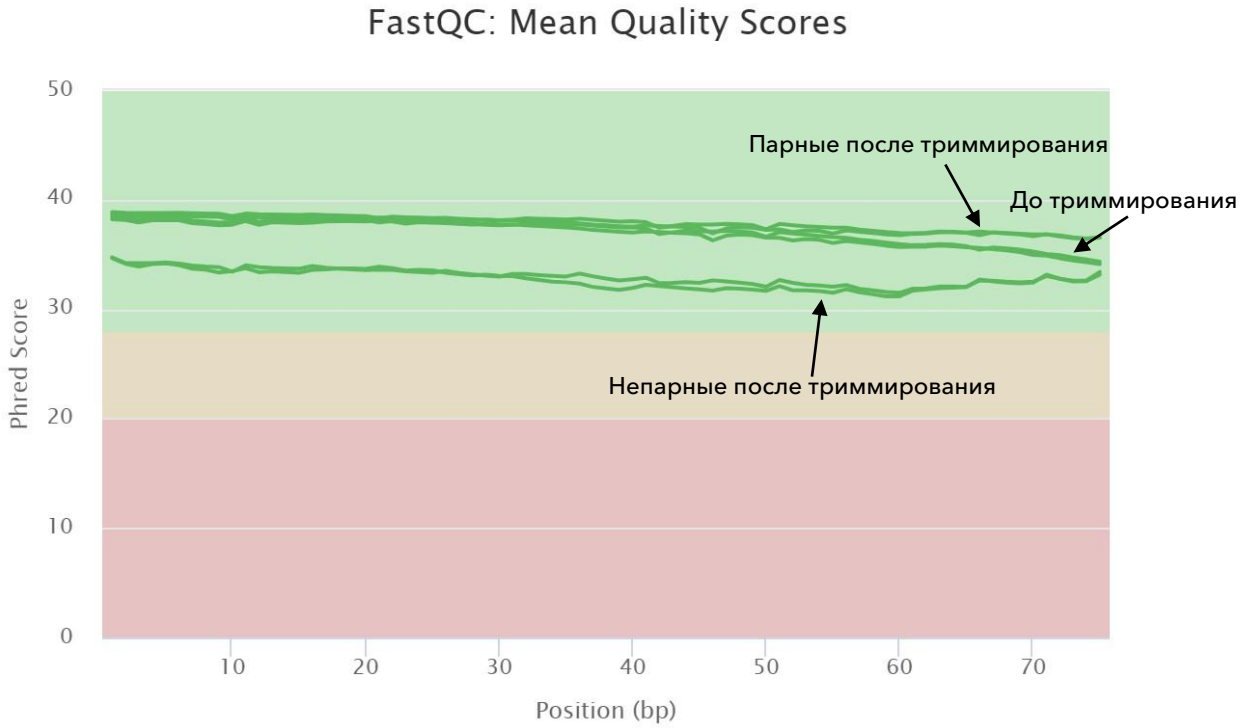


Рис. 13 Средние значения Q для каждого нуклеотида. Можно легко понять, какой паре линий соответствуют какие файлы.



Практикум 12

Картирование чтений на референсный геном

Картирование производилось при помощи программы hisat2:

```
hisat2 -x ../reference/chr12 -1
../reads/SRR10720410_1.fastq.gz -2
../reads/SRR10720410_2.fastq.gz -p 4 --no-spliced-alignment >
chr12_map.sam 2> hisat2_log.txt
```

`-x ../reference/chr12` - префикс файлов с аннотацией референса
`-1 SRR10720410_1.fastq.gz` - файл с прямыми чтениями
`-2 SRR10720410_2.fastq.gz` - файл с обратными чтениями
`-p 4` - количество ядер, на которые процесс распараллелен (честно говоря, я не знаю, как узнать, сколько их там есть и сколько надо, но нам сказали, что больше 4 всё равно выделить нельзя)
`--no-spliced-alignment` - не рассматривается возможность сплайсинга
Output: записываю в `chr12_map.sam`
Поток **error:** перенаправляю в `hisat2_log.txt`. Там содержатся информационные сообщения.

*это я читаю мануал к hisat

Конвертация sam файла в bam файл

a. Полученный sam файл весит **15 Гб**
   Это очень много, поэтому конвертируем в совершенно аналогичный bam файл:

```
samtools sort -o chr12_map.bam chr12_map.sam
```

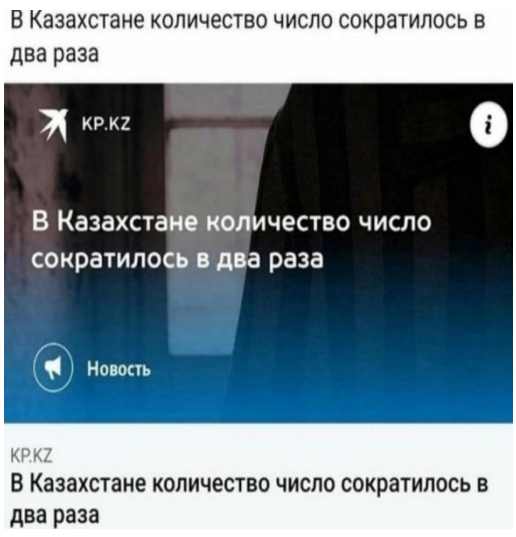
b. bam файл весит уже **4.2 Гб**

Теперь проиндексируем bam файл:

```
samtools index chr12_map.bam
```

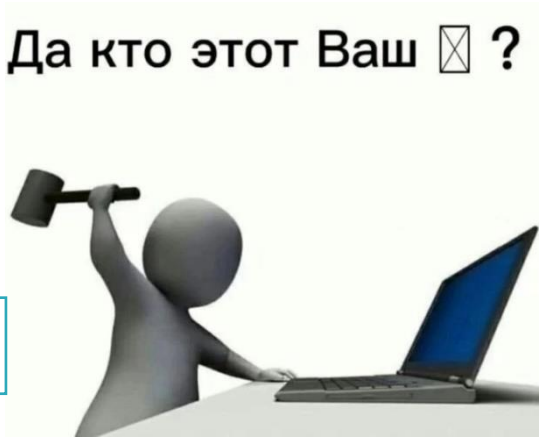
Input: chr12_map.bam
Output: chr12_map.pai

Ну а sam файл можно снести, чтобы он, такой здоровенный, не занимал память.



Анализ bam файла

bam файлы просто так просмотреть и интерпретировать нельзя: и потому, что это бинарный файл, и потому, что это куча информации.



Проанализируем его при помощи samtools flagstat (программа, которая смотрит на флаги и составляет по ним статистику)

```
samtools flagstat chr12_map.bam
> analyzed_bam.txt
```

Input: chr12_map.bam
Output: analyzed_bam.txt

Содержимое analyzed_bam.txt

```
78620423 + 0 in total (QC-passed reads + QC-failed reads)
78201516 + 0 primary
418907 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
5037547 + 0 mapped (6.41% : N/A)
4618640 + 0 primary mapped (5.91% : N/A)
78201516 + 0 paired in sequencing
39100758 + 0 read1
39100758 + 0 read2
4036660 + 0 properly paired (5.16% : N/A)
4112846 + 0 with itself and mate mapped
505794 + 0 singletons (0.65% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

a. На референс картировалось всего **5,037,547** штук чтений
b. Это **6.41%** от общего числа после триммирования
c. Из них в корректных парах картировалось **4,036,660** штук чтений
d. Это **5.16%** от общего числа после триммирования

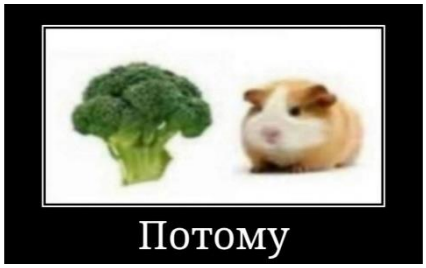
Получение чтений, картированных на 12 хромосому

«Правильное» название 12 хромосомы, как было получено ранее с помощью команды samtools faidx - **12.**

```
samtools view -h -bS chr12_map.bam 12 > out_map.bam
```

`-h` - вывод вместе с заголовком
`-b` - вывод в bam файл
`-S` - автоматическое определение типа файла ввода
Input: chr12_map.bam
Output: out_map.bam

Каким-то непонятным образом в этот момент файл chr12_map.bam оказался пустым, хотя буквально только что я его анализировала с помощью flagstat. Я буквально не могу предположить, что пошло не так, какие-то технические шоколадки.



Так что иду запускать все программы из практикума ещё раз)))

Ну как обычно, впрочем.

Получение только правильно картированных пар чтений

Теперь нужно выделить те прочтения, которые картировались правильно:

```
samtools view -f 0x2 -bS out_map.bam > correct_map
```

`-f 0x2` - те прочтения, у которых значение флага 0x2. В мануале sam файлов написано, что оно соответствует ситуации, когда оба сегмента выровнялись номально.

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Проанализируем отобранные чтения:

```
samtools flagstat correct_map.bam > analyzed_correct_bam.txt
```

Содержимое файла следующее:

```
4192232 + 0 in total (QC-passed reads + QC-failed reads)
4036660 + 0 primary
155572 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
4192232 + 0 mapped (100.00% : N/A)
4036660 + 0 primary mapped (100.00% : N/A)
4036660 + 0 paired in sequencing
2018330 + 0 read1
2018330 + 0 read2
4036660 + 0 properly paired (100.00% : N/A)
4036660 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

a. На референс картировано 4036660 штук корректных пар.
b. 100% из них картировались правильно.
c. Проиндексируем полученный bam файл, содержащий только правильно спаренные картированные чтения нужной хромосомы:

```
samtools index correct_chr7_map.bam
```

Input: correct_chr7_map.bam
Output: correct_chr7_map.bam.bai

Практикум 13

Получение вариантов

```
bcftools mpileup -f ../reference/chr12.fa
../mapping/correct_map.bam | bcftools call -mv -o variants.vcf
```

Input: `../reference/chr12.fa` - референсная последовательность,
`../mapping/correct_map.bam` - файл с картированием
Output: `variants.vcf` - файл с вариациями последовательности

`mpileup` - генерирует vcf файл, содержащий в себе информацию о правдоподобии генотипов.
`call` - отбор только нужных строчек
`-v` - только вариабельные сайты
`-m` - отбирает мультиаллельные и редкие варианты
`-o` - вывод в файл

Рис. 1 Общая схема содержимого файла vcf

«Шапка» - начинаются с ##

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##bcftoolsVersion=1.11+htslib-1.11-4
##bcftoolsCommand=mpileup -f ../reference/chr12.fa ../mapping/correct_map.bam
##reference=file://../reference/chr12.fa
##contig=<ID=12,length=133275309>
##ALT=<ID=*,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of raw reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of raw reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts in RNA-seq data (bigger is better)",Version="3">
##INFO=<ID=RPB,Number=1,Type=Float,Description="Mann-Whitney U test of Read Position Bias (bigger is better)">
##INFO=<ID=MQB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality Bias (bigger is better)">
##INFO=<ID=BQB,Number=1,Type=Float,Description="Mann-Whitney U test of Base Quality Bias (bigger is better)">
##INFO=<ID=MQSB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality vs Strand Bias (bigger is better)">
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=ICB,Number=1,Type=Float,Description="Inbreeding Coefficient Binomial test (bigger is better)">
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward and alt-reverse bases">
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.11+htslib-1.11-4
##bcftools_callCommand=call -mv -o variants.vcf; Date=Mon Dec 18 19:52:07 2023
```

Заголовки столбцов

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  ../mapping/correct_map.bam
```

Тело файла

```
12      10677   .       C       G       10.7923 .       DP=1;SGB=-0.379885;MQ0F=0;AC=2;AN=2;DP4=0,0,1,0;MQ=60   GT:PL   1/1:40,3,0
12      10919   .       G       A       6.51248 .       DP=1;SGB=-0.379885;MQ0F=0;AC=2;AN=2;DP4=0,0,0,1;MQ=60   GT:PL   1/1:35,3,0
12      11298   .       A       C       4.38466 .       DP=1;SGB=-0.379885;MQ0F=0;AC=2;AN=2;DP4=0,0,1,0;MQ=60   GT:PL   1/1:32,3,0
12      11446   .       C       T       8.99921 .       DP=1;SGB=-0.379885;MQ0F=0;AC=2;AN=2;DP4=0,0,1,0;MQ=60   GT:PL   1/1:38,3,0
12      11493   .       G       C       47.4146 .       DP=3;VDB=0.199299;SGB=-0.511536;MQ0F=0;AC=2;AN=2;DP4=0,0,3,0;MQ=60      GT:PL   1/1:77,9,0
```

В теле файла можно найти следующие столбцы:

CHROM	Имя референсной последовательности
POS	Позиция этой вариации в референсной последовательности (нумерация с 1)
ID	Идентификатор вариации («.», если неизвестен)
REF	Вариант в референсной последовательности
ALT	Список альтернативных вариантов в данной позиции
QUAL	Качество вариантов
FILTER	Обычно либо фильтр, который вариант не прошёл, либо PASS, если он прошёл все фильтры, но в данном случае столбец пустой (заполнен «.»)
INFO	Список пар вида ключ=значение, описывающих вариант
FORMAT	Формат следующего столбца
SAMPLE	Информация об образце (например, генотип по данной вариации)

vcf получился болбшой, поэтому проанализируем его содержимое:

```
bcftools stats variants.vcf > variants_stats.txt
```

Посмотрим на кусочек полученного файла:

```
# SN    [2]id   [3]key  [4]value
SN      0       number of samples:      1
SN      0       number of records:      67503
SN      0       number of no-ALTs:      0
SN      0       number of SNPs: 65525
SN      0       number of MNPs: 0
SN      0       number of indels:       1978
SN      0       number of others:       0
SN      0       number of multiallelic sites:   34
SN      0       number of multiallelic SNP sites:       34
```

a. Всего вариантов: **67503**
b. Из них **65525** - SNP
c. Количество инделей: **1978**

Фильтрация вариантов

Фильтруем варианты с нужными нам характеристиками в отдельный файл:

```
bcftools filter -i'%QUAL>30 && DP>50' variants.vcf -o
filtered_variants.vcf
```

`QUAL>30` – качество выше 30
`DP>50` – глубина прочтения этой буквы выше 50 (?)

Точно так же анализируем:

```
bcftools stats filtered_variants.vcf >
filtered_variants_stats.txt
```

Получили следующее:

```
# SN    [2]id   [3]key  [4]value
SN      0       number of samples:      1
SN      0       number of records:      2076
SN      0       number of no-ALTs:      0
SN      0       number of SNPs: 2014
SN      0       number of MNPs: 0
SN      0       number of indels:       62
SN      0       number of others:       0
SN      0       number of multiallelic sites:   6
SN      0       number of multiallelic SNP sites:       6
```

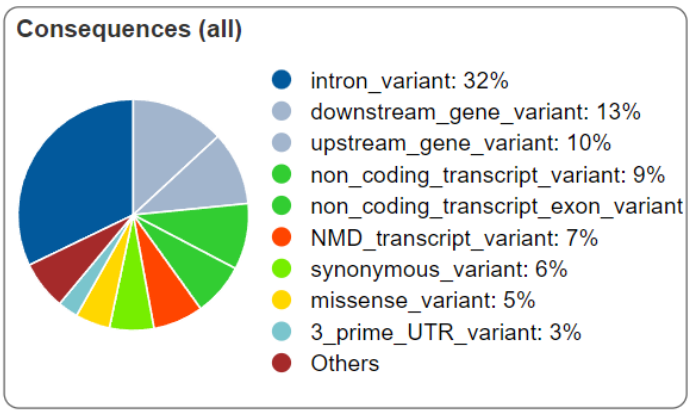
a. Осталось вариантов: **2076 (3,08%)**
b. Из них осталось **2014 (3,07%)** - SNP
c. Осталось инделей: **62 (3,13)**

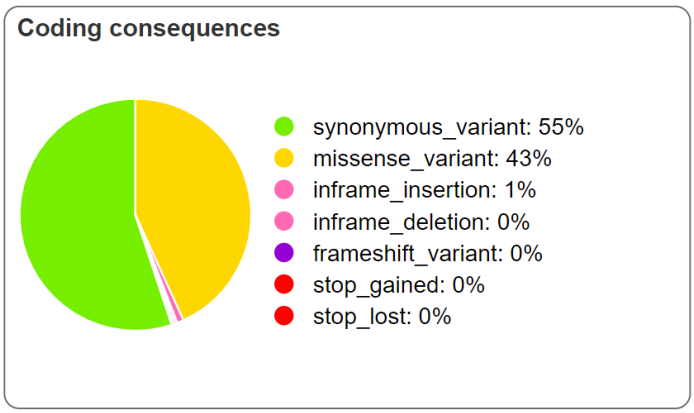
Аннотация вариантов

Аннотация отфильтрованных вариантов с помощью веб-версии VEP (variant effect predictor).

Категория	Количество
Обработанные варианты	2076
Отфильтрованные варианты	0
Новые (не зарегистрированные в базах данных) / зарегистрированные	445 (21.4) / 1631 (78.6)
В перекрывающихся генах	851
В перекрывающихся транскриптах	4574
В перекрывающихся регуляторных областях	223

Рис. 2 Эффекты вариантов: всех либо только в кодирующих последовательностях.





*NMD (nonsense-mediated mRNA decay) - нонсенс-опосредованный распад мРНК; если в мРНК не в том месте оказался стоп-кодон, она деградирует.

VEP присваивает каждому варианту одно из 4 значений параметра impact:

a. HIGH – например, сдвиг рамки считывания, появление/исчезновение стоп-кодона и т. д.
b. MODERATE – миссенс-мутации
c. LOW – синонимичные мутации
d. MODIFIER – в некодирующих участках

Всего 21 вариант имеет Impact HIGH:

- Они все попали в какой-либо ген
- Попали как в экзоны, так и в интроны
- Варианты мутаций, которые получились в данном случае:
  stop_gained/lost
  splice_donor_variant/splice_acceptor_variant
  frameshift_variant
  non_coding_transcript_variant (вместе с чем-то ещё)

Практикум 14

Описание образца

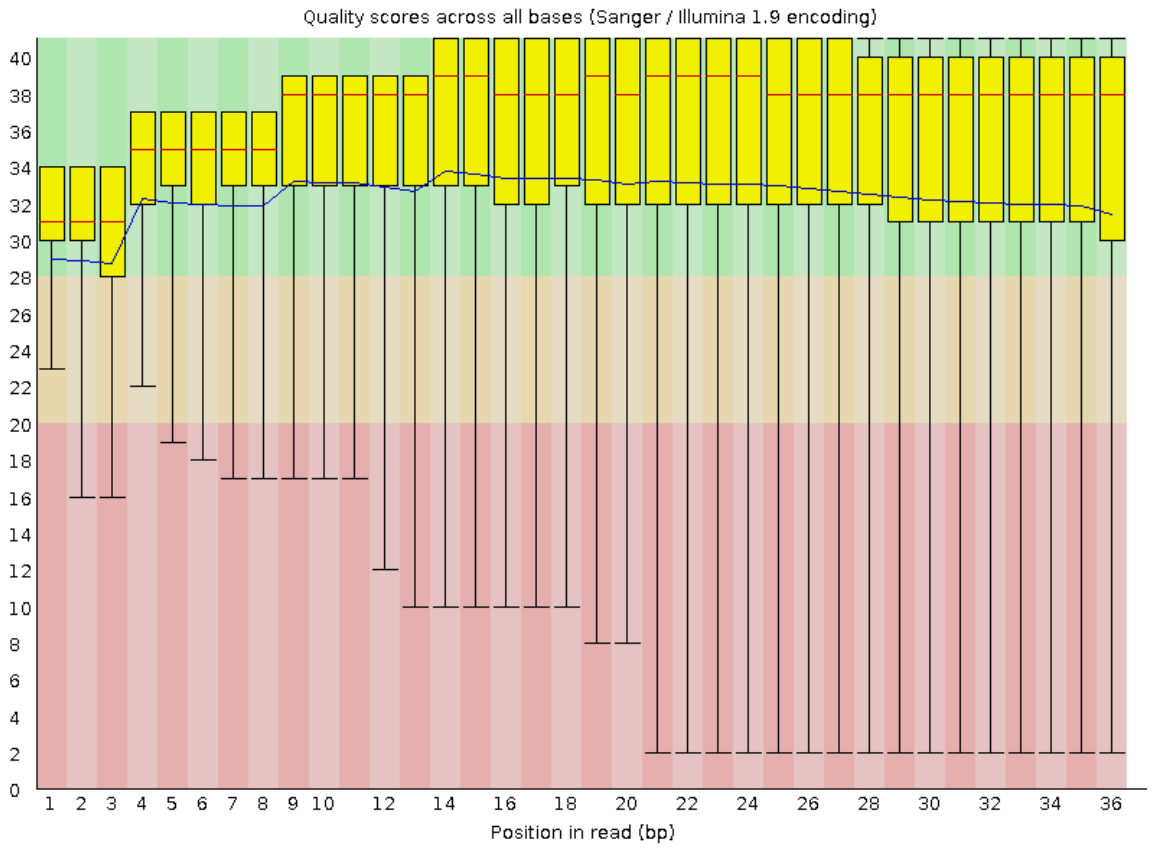
a. ID образца РНК-чтений: **ENCFF038OLY**
b. ссылка на информацию об образце: воть
c. организм и ткань: мышца ноги эмбриона человека (127 дней, мужского пола)
d. стратегия секвенирования: **polyA plus RNA-seq** (видимо, берётся вся РНК, и происходит амплификация с олигоТ праймером, в итоге получаем смесь из мРНК и немного длинных некодирующих РНК - lncRNA)
e. парноконцевые или одноконцевые чтения: **одноконцевые**
f. цепь-специфичность: нет (вроде)

Проверка качества исходных чтений

```
fastqc ENCFF038OLY.fastq.gz
```

a. количество чтений: **72,517,664**
b. качество чтений по результатам fastqc: ой…

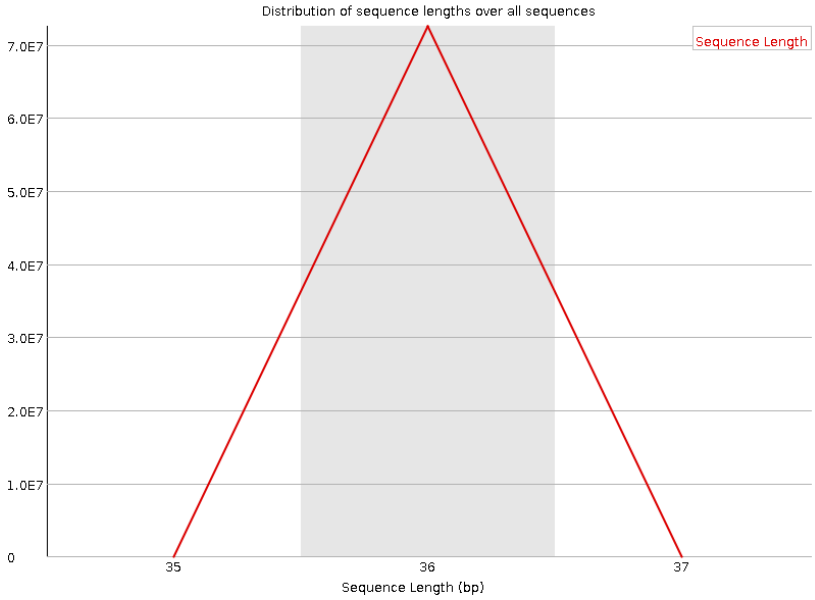
Рис. 1 Per base sequence quality; медиана обозначена красным, а среднее – синим.



Ну, по крайней мере и медиана, и среднее находятся в пределах зелёной полоски.

c. Длина чтений: все 36 нуклеотидов

Рис. 2 Распределение последовательностей по длине



d. И ещё немного данных:

Рис. 3 Качество чтений по ячейкам. Видно, что одной крайней ячейке что-то особенно грустно, но в целом, не смертельно

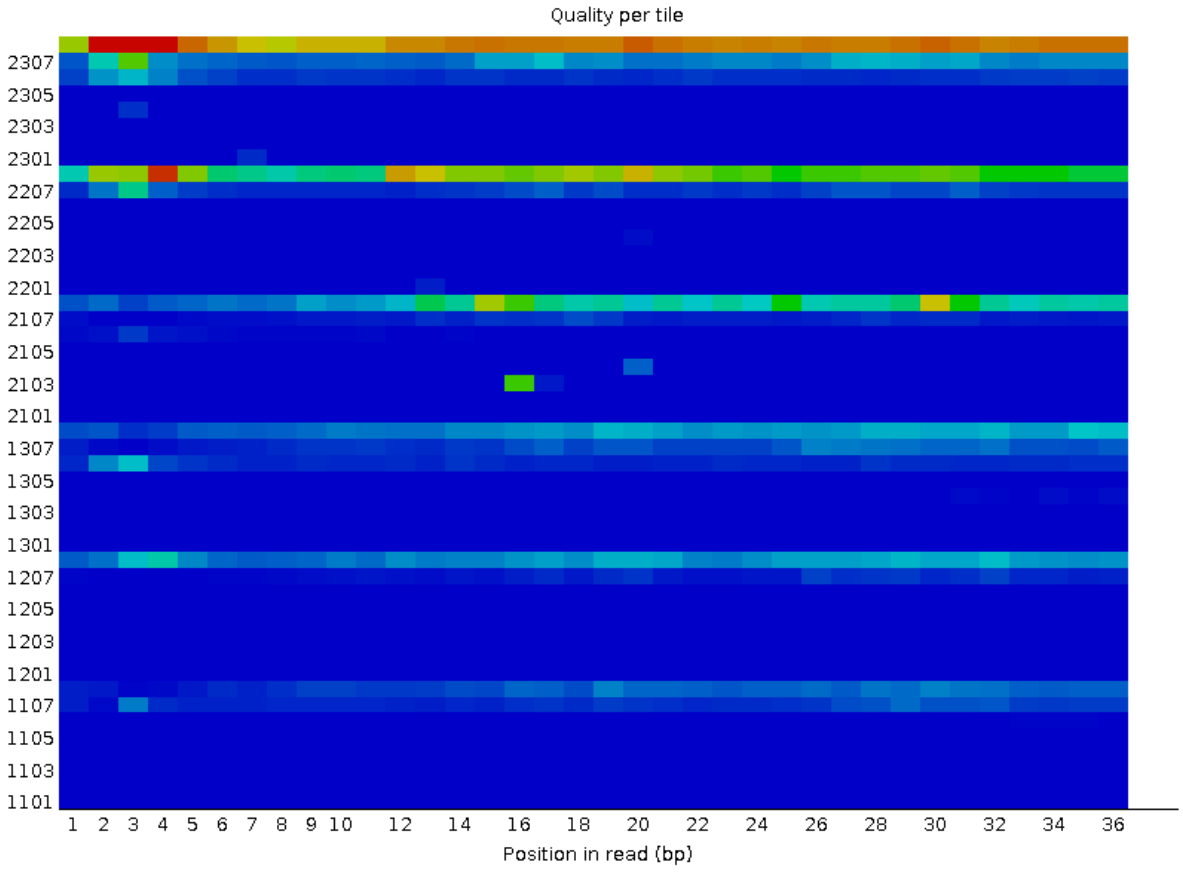


Рис. 4 Распределение последовательностей по среднему качеству. Есть пик в районе 2, что не очень хорошо

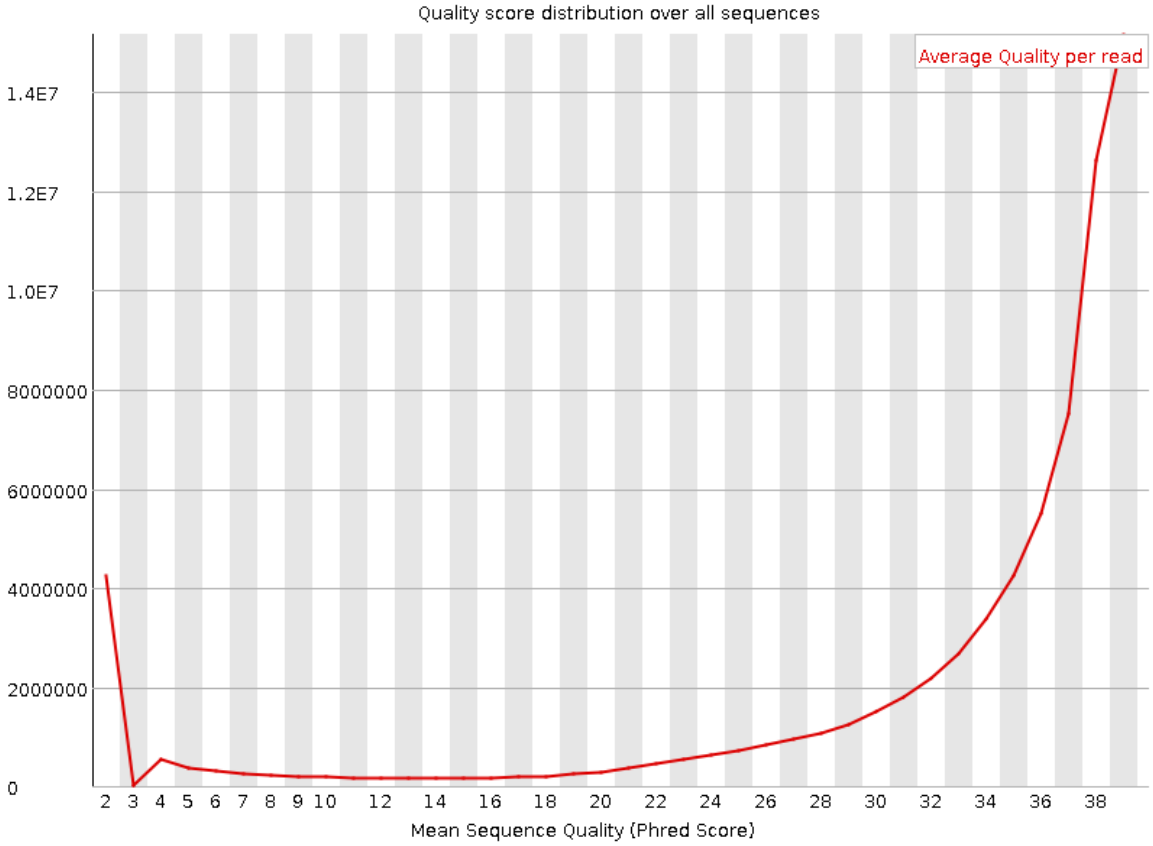


Рис. 5 Состав последовательностей (встречаемость оснований); fastqc ругается, что в начала линии такие «волнистые», и вообще такого быть не должно, по идее

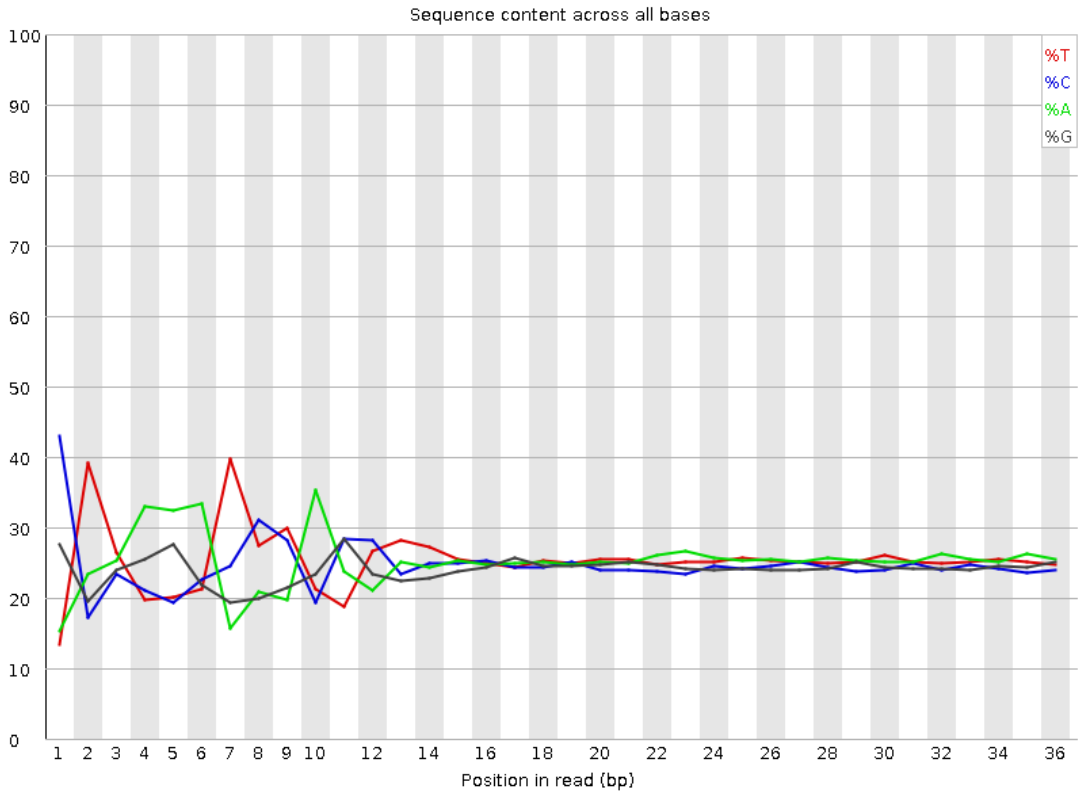


Рис. 6 Уровень дупликации последовательностей – есть довольно много последовательностей с большим количеством копий, на что fastqc тоже ругается.

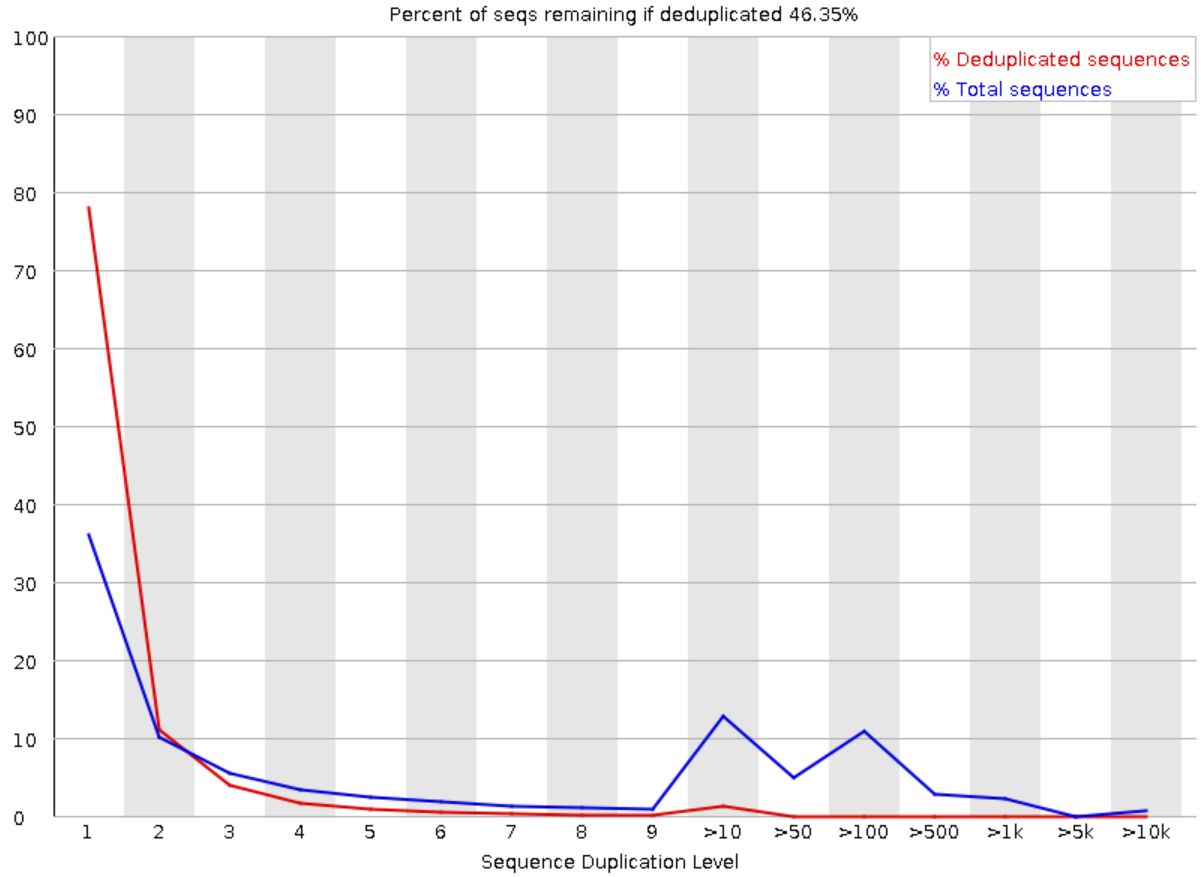


Рис. 7 Перепредстваленные последовательности – есть две таких последовательности, но они обе даже вместе составляют меньше процента от всех последовательностей, так что может быть, и ничего страшного.

Sequence	Count	Percentage	Possible Source
GATCGGAAGAGCACACGTCTGAACTCCAGTCACTAG	459179	0.6331960720632148	TruSeq Adapter, Index 7 (97% over 36bp)
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACTA	145109	0.2001015917997579	TruSeq Adapter, Index 7 (97% over 35bp)

Картирование чтений на референс

Референс тот же - 12 хромосома, и он уже проиндексирован. Картируем чтения на него с помощью hisat2:

```
hisat2 -x ../../reference/chr12 -k 3 -U
../reads/ENCFF038OLY.fastq.gz > rna_map.sam 2> rna_map_log.txt
```

`-x ../../reference/chr12` - префикс файлов с проиндексированным референсом.
`-k 3` - количество лучших первичных выравниваний, которые отбирает hisat2. Из-за того, что это лишь первичная оценка, нет гарантии, что это вообще лучшие выравнивания.
`-U ../reads/ENCFF038OLY.fastq.gz` - список файлов или, в данном случае, один файл с чтениями

Output: `rna_map.sam`
Stderr: перенаправляется в `rna_map_log.txt`

Посмотрим на лог-файл:

```
72517664 reads; of these:
  72517664 (100.00%) were unpaired; of these:
    66283649 (91.40%) aligned 0 times
    5227783 (7.21%) aligned exactly 1 time
    1006232 (1.39%) aligned >1 times
8.60% overall alignment rate
```

a. На 12 хромосому картировалось **6,234,015** чтений (**8,60%** от общего количества), что даже удивительно много, потому что в данных секвенирования были РНК всякие разные, а референс – только одна хромосома.

Переводим sam файл в сортированный bam:

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

Индексируем bam файл:

```
samtools index rna_map.bam
```

Отбираем только чтения, которые картировались на 12 хромосому:

```
samtools view -h -bS rna_map.bam 12 > chr12_map.bam
```

Поиск экспрессирующихся генов

Есть файл с разметкой данной хромосомы, рассмотрим его содержимое:

```
##description: evidence-based annotation of the human genome, version 27
(Ensembl 90), mapped to GRCh37 with gencode-backmap - basic transcripts
##provider: GENCODE
##contact: gencode-help@sanger.ac.uk
##format: gtf
##date: 2017-08-01
chr12   HAVANA  gene    67607   69138   .       +       .       gene_id
"ENSG00000249054.2_2"; gene_type "lincRNA"; gene_name "FAM138D"; level 2;
havana_gene     "OTTHUMG00000167962.2_2";       remap_status    "full_contig";
remap_num_mappings 1; remap_target_status "overlap";
```

У файла есть шапка, в которой каждая строка начинается с ##, и в ней содержится основная информация об этой аннотации в целом (дата, что и как аннотировали, к кому обращаться с претензиями в случае чего).

Далее – огромное количество строчек, в каждой строчке информация о штуке, которую мы аннотируем. Информация разделена на следующие столбцы, разделённые табами:

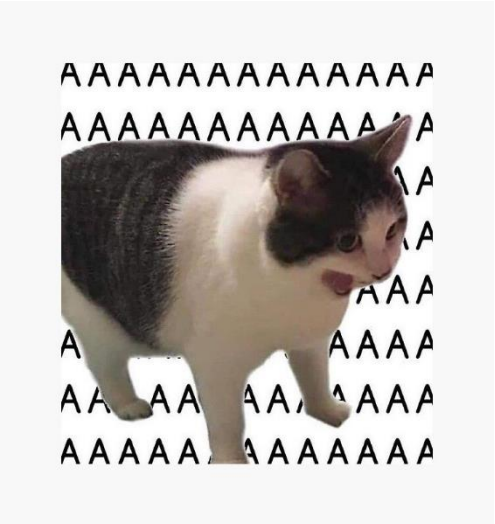
Название столбца	Содержимое	Пример из файла для 12 хромосомы
seqname	Название хромосомы/скаффолда	Chr12
source	Название программы, которая создала эту аннотацию, либо источник данных (например, база данных)	HAVANA
feature	Тип аннотации	gene
start	Координата начала (нумерация с 1)	67607
end	Координата конца (нумерация с 1)	69138

score	Вещественное число, которое обозначает  ? Видимо, достоверность аннотации или её качество	. (неизвестно, но пустое место нельзя оставлять)
strand	«+» или «-»	+
frame	0, 1 или 2 - рамка считывания (для генов)	.
attribute	Список пар тэг-значение, разделённых «;»	gene_id "ENSG00000249054.2_2"; gene_type "lincRNA"; gene_name "FAM138D"; …

Хотим узнать, сколько генов аннотировано на 12 хромосоме, то есть сколько есть записей, у которых в 3 столбце написано «gene»:

```
tail -n +5 gencode.hg19.v27.chr12.gtf | cut -f3 | grep 'gene' | wc -l
```

Эта строчка выводит число **3063.**
В то же время в интернете пишут максимум про «больше 1600»...



Теперь нужно посчитать, сколько чтений картировалось на каждый ген:

```
htseq-count -f bam -s no -t gene -o out_genes.sam chr12_map.bam ../annotation/gencode.hg19.v27.chr12.gtf > count_gene.txt 2> count_log.txt
```

`-f` (от format) – формат вводных данных – sam/bam. По умолчанию стоит sam.
`-s` (stranded) – есть ли в методе цепь-специфичность; по умолчанию – yes.
`-m` (mode) – правила определения множества аннотаций, на которые картировалось чтение; по умолчанию **union.**
`-t` (type) – берём только аннотации с указанным здесь значением в 3 столбце.
`-o` (samout) – вывести все записи о выбавниваниях в виде sam файла с пометкой, на какую аннотацию он картировался

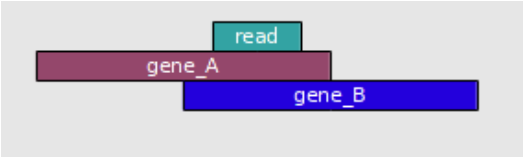
`count_gene.txt` – текстовая выдача
`count_log.txt` – log-файл, сюда перенаправлен stderr

В log-файле не нашла ничего особо интересного, разве что общее количество обработанных выравниваний – **7,820,413**.

Конец файла count_gene.txt выглядит следующим образом:

```
__no_feature    5227783
__ambiguous     0
__too_low_aQual 0
__not_aligned   0
__alignment_not_unique  1006232
```

__no_feature - множество генов, на которые мы определили, что чтение картировалось, пустое.
__ambiguous - возникла такая ситация:



И не очень понятно, что писать: gene_A или gene_B.
__too_low_aQual - если указана опция -a, можно установить порог качества выравнивания, и тогда чтения с выравниваниями ниже порогового посчитаются здесь.
__not_aligned - в bam файле не было ни одного выравнивания для этого чтения
__alignment_not_unique - в bam файле было несколько выравниваний в разные места

Для полного счастья не хватает только посчитать те чтения, которые выровнялись на какой-то ген однозначным образом:

```
path = r"C:\Users\suvor\Documents\count_gene.txt"

count = 0
with open(path) as file:
    for line in file:
        if line.startswith('_'):
            break
        count += int(line.split()[1])
print(count)
```

```
0
```

А это уже что-то странное... Я просмотрела файл, там правда везде 0, но это буквально не имеет никакого смысла.

В границы генов попало 1,006,232 + 0 = **1,006, 232** чтений.
Мимо границ генов попало **5,227,783** чтений.