Сборка и анализ геномов

Петренко П. С.

Факультет биоинженерии и биоинформатики, Московский Государственный Университет имени М.В.Ломоносова
Контактные данные: Paull_p_s@mail.ru

Практикум 11.

Введение в анализ NGS.

Часть 1.

Задача практикума: подготовить необходимые файлы (парно-концевые прочтения и последовательность референсного генома), изучить качество предложенных чтений, проиндексировать референс.

1. Подготовка референса.

1) Получение референса:

Мне досталась 14 хромосома человека. Эта хромосома содержит примерно 107 млн пар оснований, что составляет от 3 до 3,5% всего материала ДНК человеческой клетки. Данные по количеству генов на хромосоме в целом разнятся из-за различных подходов к подсчёту. Вероятно, она содержит от 700 до 1300 генов. Некоторые из генов, расположенных на 14-й хромосоме, включают RNR2 (рибосомная РНК 2), HIF-1α (ген α-субъединицы фактора, индуцируемого гипоксией 1), и TSHR (рецептор тиреотропного гормона).

Для начала с помощью функции hisat2 проиндексируем хромосому, чтобы создать карту структуры данных последовательности хромосомы для лучшей работы программ при дальнейшем картировании:
`hisat2-build Homo_sapiens.GRCh38.dna.chromosome.14.fa chr14_indexed`
`chr14_indexed` - индекс, обеспечивающий произвольный доступ к файлам FASTA
Получили 8 файлов с названиями: `chr14_indexed.?.ht2`, где ? - это число от 1 до 8.

2) Индексация samtools:

Теперь проиндексируем хромосому с помощью samtools, так мы получим файл-индекс, который поможет быстро перейти к нужной позиции в файле:
`samtools faidx Homo_sapiens.GRCh38.dna.chromosome.14.fa`
`faidx` - так будут начинаться индексные файлы

Получили файл `Homo_sapiens.GRCh38.dna.chromosome.14.fa.fai` со следующей информацией:

Таблица 1. Данные samtools индексации.

NAME	LENGTH	OFFSET	LINEBASES	LINEWIDTH
Название последовательности (имя хромосомы)	Длина последовательности (в нуклеотидах)	Смещение до первого нуклеотида (в байтах) - по сути это название	Количество нуклеотидов в каждой строке	Вес строки в байтах (с учётом переноса строки)
14	107043718	58	60	61

1. Чтения ДНК.

1) Описание образца:

Таблица 2. Описание образца с помощью базы NCBI.

SRR ID образца ДНК-чтений	SRR10720407
Ссылка на информацию об образце из NCBI	Ex CT-12N
Прибор для секвенирования	Illumina Genome Analyzer IIx
Организм	Homo sapiens
Стратегия секвенирования	Whole-exome (цельно-экзомная) (в другом поле OTHER)
Тип чтений	PAIRED (парноконцевые)
Ожидаемое количество чтений	38,530,707

2) Проверка качества исходных чтений:

Проведём анализ исходных данных секвенирования, чтобы выявить проблемы с данными сразу, если они есть, и избежать некорректных результатов:

```
fastqc SRR10720407_1.fastq.gz
fastqc SRR10720407_2.fastq.gz
```

Получили два html файла (`SRR10720407_1_fastqc.html и SRR10720407_2_fastqc.html`) и два zip файла (`SRR10720407_1_fastqc.zip и SRR10720407_1_fastqc.zip`).

a) Получилось `38530707 пар чтений`

b) Количество чтений у “прямых” чтений и “обратных” чтений совпадает

c) Качество пар чтений можно оценить как хорошее, все боксплоты находятся в зелёной зоне (медиана и среднее значение), только у четырёх боксплотов прямого прочтения и шести боксплотов обратного прочтения усы выходят в жёлтую зону (рис.1 и 2).

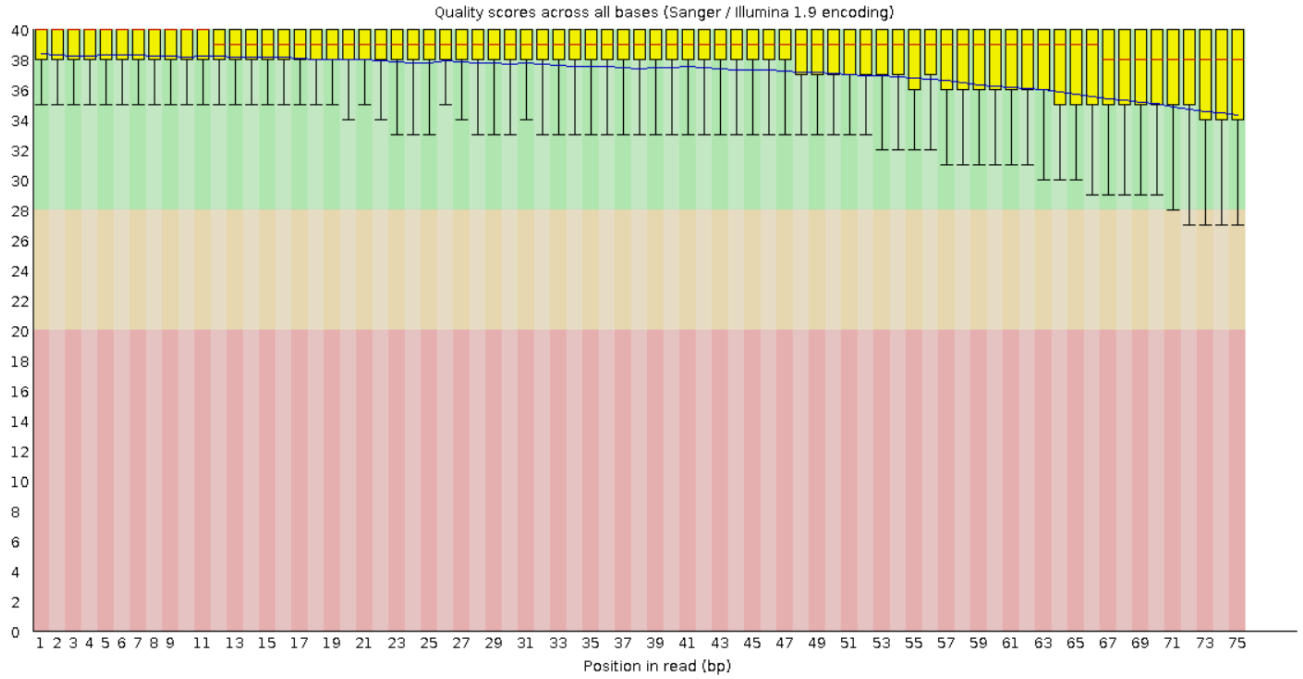

Рис. 1. Картинка Per base sequence quality для прямого прочтения.

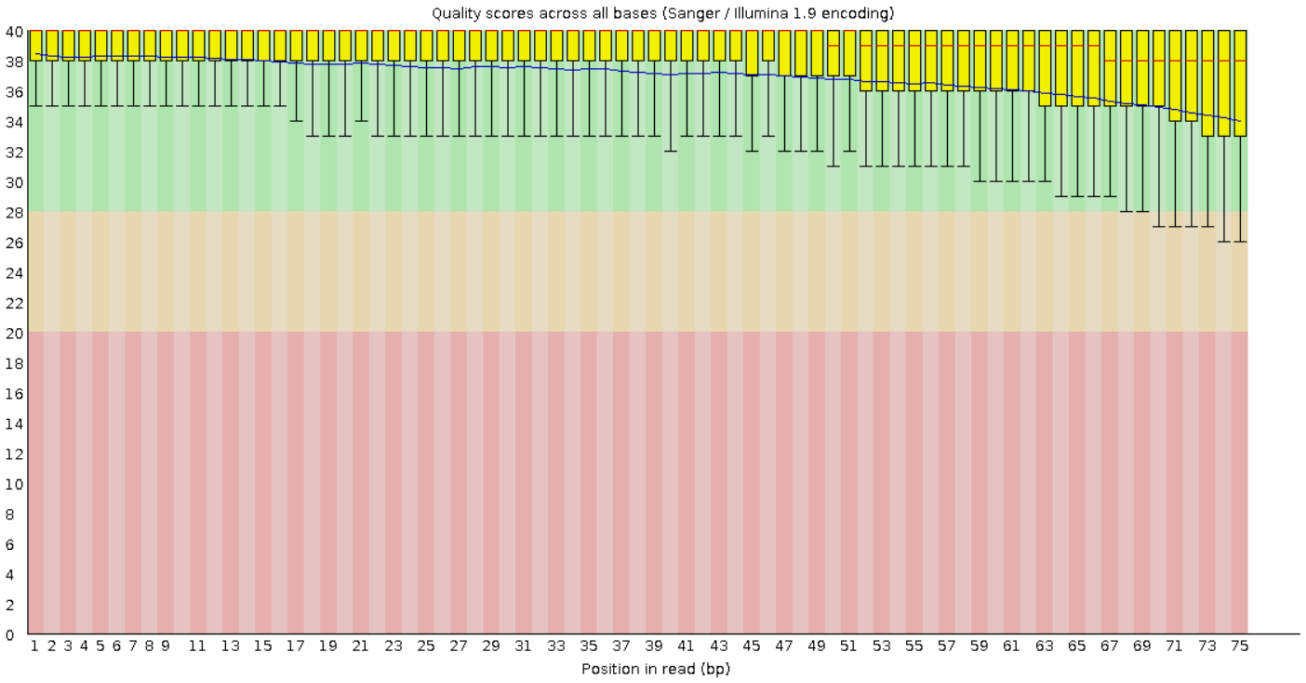

Рис. 2. Картинка Per base sequence quality для обратного прочтения.

d) Длина чтений прямых и обратных равна 75 нуклеотидам (рис. 3 и 4).

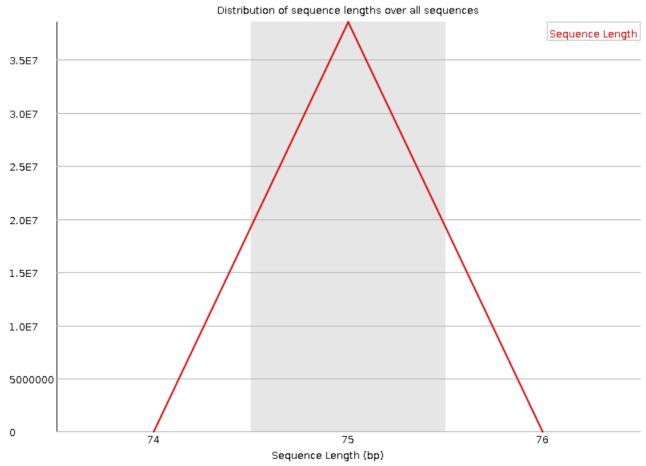

Рис. 3. Картинка Sequence Length Distribution для прямого прочтения.

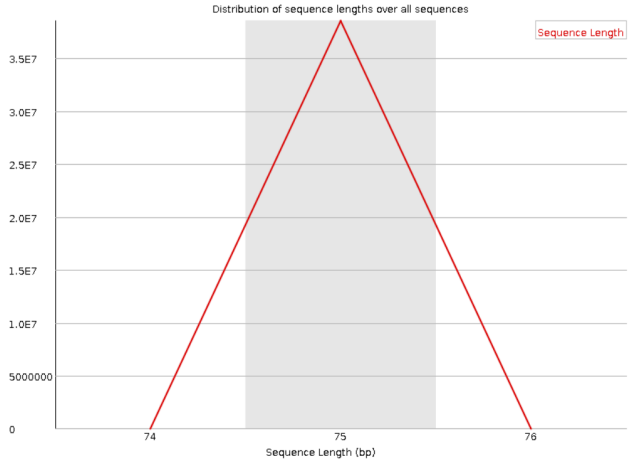

Рис. 4. Картинка Sequence Length Distribution для обратного прочтения.

3) Фильтрация чтений:

Отфильтруем чтения, чтобы удалить низкокачественные нуклеотиды (с качеством ниже 20) и всякие артефакты секвенирования. Используем trimmomatic с режимом PE для парноконцевых прочтений:

```
TrimmomaticPE -phred33 SRR10720407_1.fastq.gz SRR10720407_2.fastq.gz SRR10720407_1_paired.fastq.gz SRR10720407_1_unpaired.fastq.gz SRR10720407_2_paired.fastq.gz SRR10720407_2_unpaired.fastq.gz TRAILING:20 MINLEN:50
```

`TRAILING:20` – удаляет с конца нуклеотиды с качеством ниже 20

`MINLEN:50` – удаляет чтения с длиной меньше 50

`-phred33` - данный Quality Score (для Illumina)

Получили 4 файла (два paired и два unpaired): `SRR10720407_1_paired.fastq.gz`, `SRR10720407_1_unpaired.fastq.gz`, `SRR10720407_2_paired.fastq.gz`, `SRR10720407_2_unpaired.fastq.gz`

p.s. paired значит, что после отбора по заданным критериям сохранились оба чтения, unpaired значит, что после отбора по заданным критериям сохранилось только одно прочтение.

4) Проверка качества триммированных чтений

Проанализируем качество чтений после обработки программой Trimmomatic с помощью программы fastQC:

```
fastqc SRR10720407_1_paired.fastq.gz
fastqc SRR10720407_2_paired.fastq.gz
fastqc SRR10720407_1_unpaired.fastq.gz
fastqc SRR10720407_2_unpaired.fastq.gz
```

Получили 4 html файла, перекинули в папочку, открыли и анализируем:

a) 37276728 пар чтений осталось (paired), совпадает с обратным прочтением.

b) 96,75% пар чтений осталось (paired), совпадает с обратным прочтением.

c) Хорошо видно, что качество paired (рис. 5) гораздо лучше, чем качество unpaired (рис. 6) (в принципе как и ожидалось, ведь по непарным фрагментам собрать целую картинку будет гораздо сложнее). В paired все боксплоты и их хвосты находятся в зелёной зоне, в то время как в unpaired часть боксплотов выходит в жёлтую зону, а 59-62 усы в прямом прочтении заходят на красную зону.

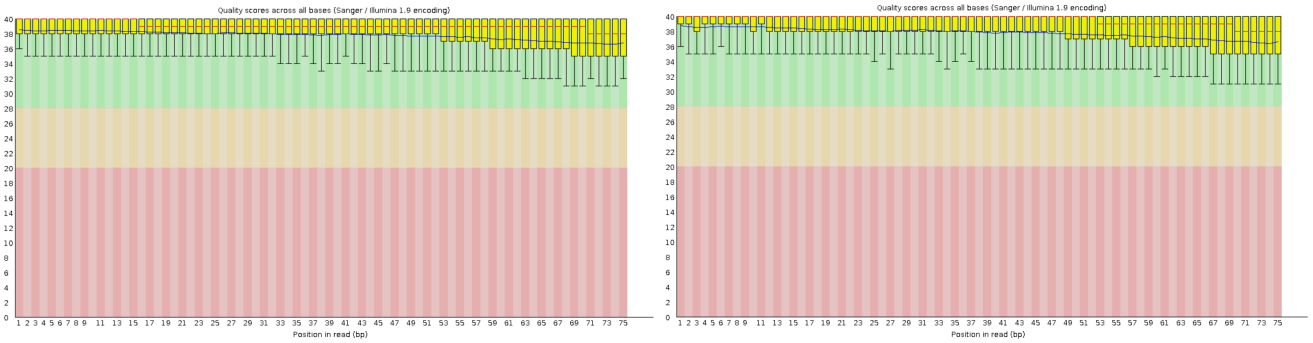

Рис. 5. Per base sequence quality для paired после триммирования (слева для прямых чтений, справа для обратных).

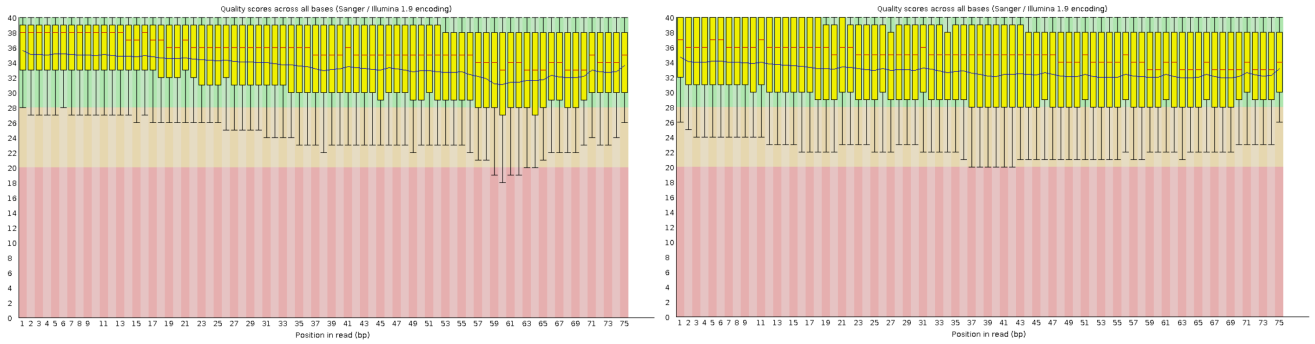

Рис. 6. Per base sequence quality для unpaired после триммирования (слева для прямых чтений, справа для обратных).

d) Видно, что после триммирования (paired) (рис. 8) качество становится лучше. Так, усы боксплотов больше не заходят на жёлтые области, а полностью находятся в зелёной области вместе с боксплотами. Это и ожидалось увидеть, так как после триммирования нам больше не мешают низкокачественные нуклеотиды (с качеством ниже 20) и артефакты секвенирования.

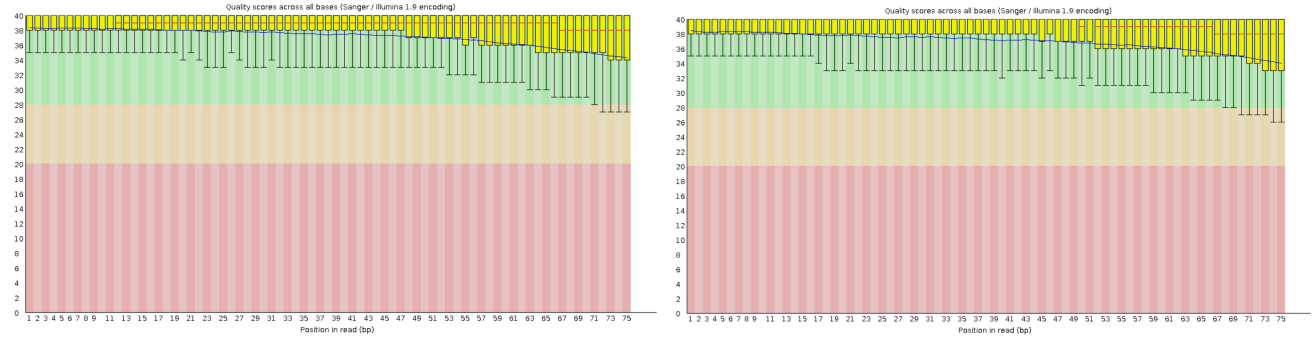

Рис. 7. Per base sequence quality до триммирования (слева для прямых чтений, справа для обратных).

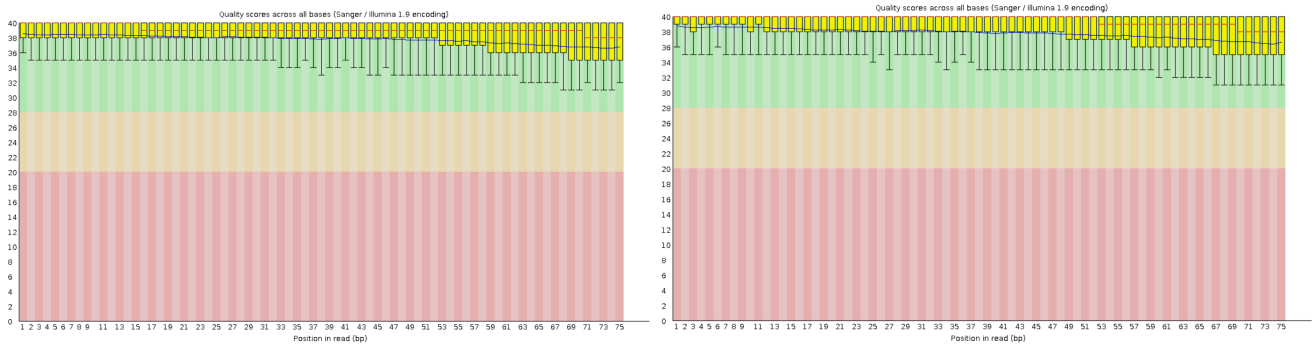

Рис. 8. Per base sequence quality после триммирования (слева для прямых чтений, справа для обратных).

e) Проанализируем длину чтений. Сначала посмотрим на прямое прочтение (рис. 9). Видим, что в paired длина осталась равной 75 нуклеотидам, тогда как в unpaired уменьшилось количество чтений с длиной 75 нуклеотидов, но при этом появились заметные прочтения в 50, 55, 60, 65 и 70 нуклеотидов. Теперь посмотрим обратное прочтение (рис.10). Здесь ситуация похожа: заметно, что в paired сохоаняется длина чтений в 75 нуклеотидов, тогда как в unpaired встречаются чтения в 50, 55, 60, 65 и 70 нуклеотидов, но их заметно меньше, чем в прямом чтении.

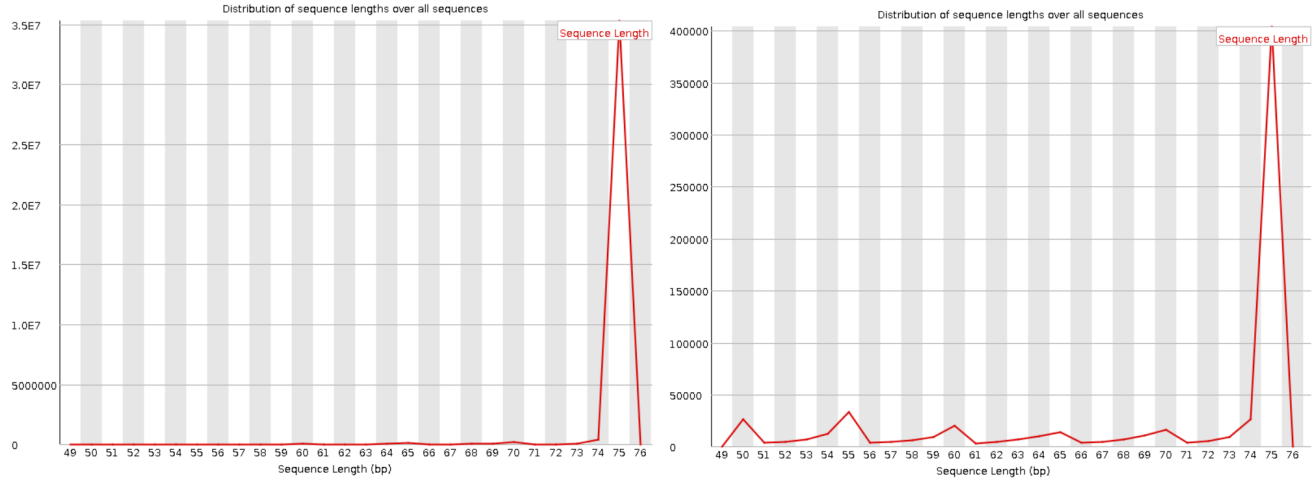

Рис. 9. Sequence Length Distribution для прямых чтений (слева для paired, справа для unpaired).

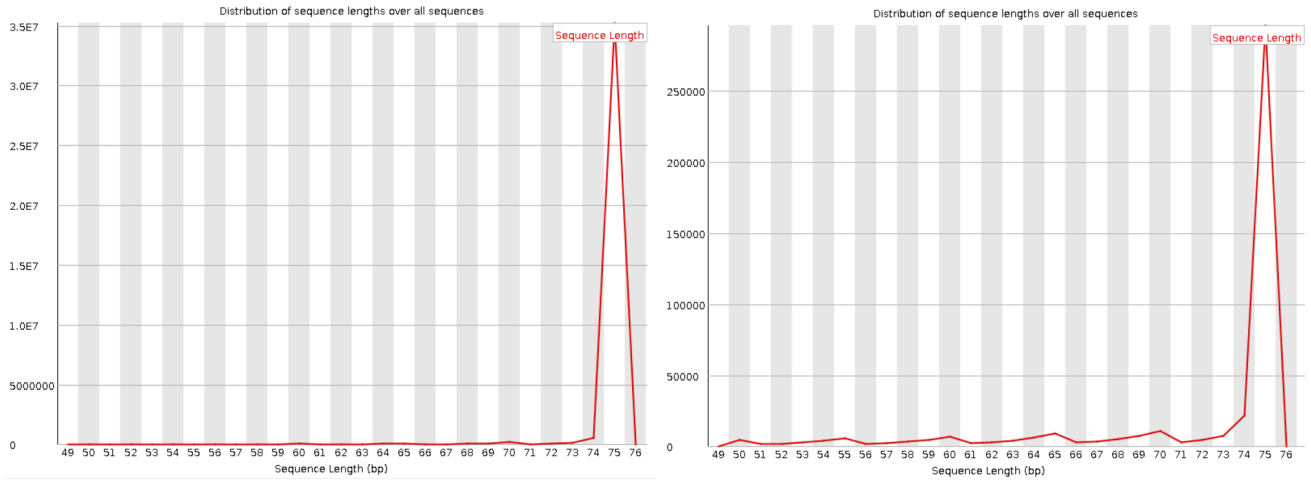

Рис. 10. Sequence Length Distribution для обратных чтений (слева для paired, справа для unpaired).

Часть 2.

Задача практикума: картировать чтения хорошего качества на референсный геном и отобрать только такие чтения, которые удалось картировать в корректных парах.

1. Картирование чтений на референсный геном.

Чтобы картировать наши хорошие чтения на референсный геном, воспользуемся программой hisat2 с нужными параметрами (будем работать в директории cart, поэтому перенесём туда индексированный референс):

```
hisat2 -x chr14_indexed -1 ../DNA_read/SRR10720407_1_paired.fastq.gz
-2 ../DNA_read/SRR10720407_2_paired.fastq.gz -p 8 --no-spliced-alignment -S mapped.sam 2>
cartlog.txt
```

`-x chr14_indexed` - префикс имен файлов с индексацией референса, которые до этого были получены с помощью hisat2-build

`-1 ../DNA_read/SRR10720407_1_paired.fastq.gz` - файл с прямыми парными триммированными чтениями

`-2 ../DNA_read/SRR10720407_2_paired.fastq.gz` - файл с обратными парными триммированными чтениями

`-p 8` – использую 8 ядер процессора

`--no-spliced-alignment` - параметр, запрещающий возможность сплайсинга

`> mapped.sam` – запись вывода команды в sam-файл

`2> cartlog.txt` – перенаправление ошибок в файл

2. Конвертация sam в bam.

1) Описание sam/bam файла:

Взвесим sam-файл: `du -h mapped.sam`

Sam-файл весит 15 Гб.Так как Sam файл очень тяжелый, переконвертируем его в сортированный bam файл и удалим sam:

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

`samtools sort` – сортирует sam-файл

`-o mapped.bam` – вывод программы в файл mapped.bam

`mapped.sam` – изначальный sam-файл

`mapped.bam` – полученный bam-файл

Взвесим sam-файл: `du -h mapped.bam`

Bam-файл весит 4 Гб.

2) Проиндексируем получившийся bam файл:

Индексация поможет быстро получить доступ к любому участку файла:

```
samtools index mapped.bam
```

`samtools index` – программа, индексирующая файл

`cart.bam` – исходный файл

Получили файл `mapped.bam.bai`.

3. Анализ bam файла.

Так как bam файл бинарный, то так просто его открыть не получится, поэтому воспользуемся программой samtools flagstat:

```
samtools flagstat mapped.bam > mapped.txt
```

`mapped.bam` – входной bam файл

`mapped.txt` – полученный txt файл

Открыли текстовый файл и изучили его:

```
75381506 + 0 in total (QC-passed reads + QC-failed reads
74553456 + 0 primary
828050 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
4175145 + 0 mapped (5.54% : N/A)
3347095 + 0 primary mapped (4.49% : N/A)
74553456 + 0 paired in sequencing
37276728 + 0 read1
37276728 + 0 read2
2637344 + 0 properly paired (3.54% : N/A)
2774156 + 0 with itself and mate mapped
572939 + 0 singletons (0.77% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5
```

С помощью некоторого сайта, на котором есть описания строк в полученном файле, постараемся ответить на заданные вопросы:

1) В поле “in total” указано количество чтений которые прошли или не прошли проверку качества: 75381506 + 0 (QC-passed reads + QC-failed reads), при этом все они успешные.

2) На картирование поступило 4175145 чтений (поле mapped).

3) 2637344 чтений картировано на референс в корректных парах (поле properly paired).

4) 3.54% чтений картировано на референс в корректных парах в процентах относительно потупивших на картирование (поле properly paired) (процент такой низкий, потому что картирование всего генома проводилось на нашу 14 хромосому).

4. Получение чтений, картированных на хромосому.

Получим чтения, картированные только на мою хромосому с помощью программы samtools view:

```
samtools view -h -bS  mapped.bam 14 > mapped14.bam
```

`samtools view` – печатает все чтения картированные на референс

`-h` – выводит в файл вместе с заголовком

`-b` – вывод в bam-файл

`-S` – формат файла в input определяет автоматически

`14` – имя моей хромосомы (посмотрел в практикуме 11)

`mapped14.bam` – полученный bam-файл с чтениями, картированными на хромосому 14

Далее с помощью samtools flagstat преобразуем bam файл в подходящий вид для анализа (как делали раньше):

```
samtools flagstat mapped14.bam > mapped14.txt
```

Открыли текстовый файл и изучили его:

```
4748084 + 0 in total (QC-passed reads + QC-failed reads)
3920034 + 0 primary
828050 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
4175145 + 0 mapped (87.93% : N/A)
3347095 + 0 primary mapped (85.38% : N/A)
3920034 + 0 paired in sequencing
1960017 + 0 read1
1960017 + 0 read2
2637344 + 0 properly paired (67.28% : N/A)
2774156 + 0 with itself and mate mapped
572939 + 0 singletons (14.62% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5
```

Посмотрим, что изменилось по сравнению с предыдущим пунктом:
1) Общее количество прочтений уменьшилось с 75381506 до 4748084 чтений.
2) 67.28% чтений картировано на референс в корректных парах, в отличие от предыдущих 3.54%.
3) Процент выравнивания повысился с 5.54% до 87.93%.
4) Но также повысился процент одиночных чтений, потерявших свою пару с 0.77% до 14.62% .

5. Получение только правильно картированных пар чтений.

Чтобы получить только правильно картированные чтения воспользуемся командой samtools view:
`samtools view -f 2 -bS mapped14.bam > cormapped14.bam`
`-f 2` – выводит в output только те чтения, которые прошли по критерию FLAG со значением 2: это значение соответствует PROPER_PAIR, то есть выведутся только чтения, которые точно выравнены с референсом
`-b` – вывод в файл формата bam
`-S` – формат файла в input определяет автоматически

Далее с помощью samtools flagstat преобразуем bam файл в подходящий вид для анализа (как делали раньше):
`samtools flagstat cormapped14.bam > cormapped14.txt`

Изучим полученный файл и сравним с файлом из пункта 4:
```
2969802 + 0 in total (QC-passed reads + QC-failed reads)
2637344 + 0 primary
332458 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
2969802 + 0 mapped (100.00% : N/A)
2637344 + 0 primary mapped (100.00% : N/A)
2637344 + 0 paired in sequencing
1318672 + 0 read1
1318672 + 0 read2
2637344 + 0 properly paired (100.00% : N/A)
2637344 + 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)
```

Общее количество прочтений опять уменьшилось с 4748084 до 2969802 чтений. Все чтения (100.00%) картированы на референс в корректных парах и процент выравнивания соотвественно стал равен 100%.

Проиндексируем файл с хорошо картированными прочтениями:
`samtools index cormapped14.bam`
Получили файл `cormapped14.bam.bai`.

6. Получение чтений, картированных только в границы экзома.

Оставим только такие чтения, которые картировались в пределах экзома средствами bedtools intersect (то есть уберём чтёния с интронами):
`bedtools intersect -abam cormapped14.bam -b /mnt/scratch/NGS/DATA/genes/seqcap_hg38.bed > exom14.bam`
`-abam` – подаёт bam-файл для сравнения на вход
`-b` – задает файл bed формата с признаками для сравнения на вход
`exom14.bam` – полученный файл работы программы, содержащий пересечение наших чтений с последовательностью хромосомы (по координатам)

Далее для полученного bam файла опять применяем samtools flagstat:
`samtools flagstat exom14.bam > exom14.txt`

Читаем полученный файл:
```
1370035 + 0 in total (QC-passed reads + QC-failed reads)
1332096 + 0 primary
37939 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
```

```
1370035 + 0 mapped (100.00% : N/A)
1332096 + 0 primary mapped (100.00% : N/A)
1332096 + 0 paired in sequencing
665464 + 0 read1
666632 + 0 read2
1332096 + 0 properly paired (100.00% : N/A)
1332096 + 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)
```

Видим, что количество экзомных правильно картированных чтений 14 хромосомы равно 1370035, что составляет 46.13% от всех правильно картированных чтений 14 хромосомы.

Практикум 12.

Анализ экзомов.

Задача практикума: получить список вариантов на основании полученного ранее bam файла и аннотировать их средствами VEP.

1. Получение вариантов

Задания этого практикума будут выполняться в директории /mnt/scratch/NGS/paull/pr12. Получим варианты и их вероятности с помощью программы bcftools mpileup:

```
bcftools mpileup -f Homo_sapiens.GRCh38.dna.chromosome.14.fa cormapped14.bam |bcftools call -mv -o variants.vcf
```

`bcftools mpileup` – генерирует vcf файл с вероятностями разных вариантов, основанных на выравнивании
`-f` – указывает референс и файл bam с картированными чтениями
`bcftools call` – из stdout программы bcftools mpileup берет только нужные строки, которые задаются опциями
`-m` – модель, которая ищет мультиаллельные и редкие варианты
`-v` – в выдаче будут только варианты
`-o` – выдача в файл (variants.vcf)

Получен файл `variants.vcf` (я его не хочу сюда вставлять скриншотом, потому что всё очень мелко и непонятно, могу открыть на колке, если надо будет). Опишем его:
Сначала идет шапка файла, каждая строка начинается с ##.
Далее идет строка с названиями столбцов.
Следующие строки – “тело” файла разбитое по столбцам.

Значения названий столбцов:
CHROM – имя хромосомы
POS – позиция варианта
ID – любая информация о варианте (у нас везде стоят точки)
ALT – альтернативный аллель
QUAL – качество варианта
FILTER – качество варианта (везде стоят точки , так как ввели файл, который уже был маркирован по качеству)
INFO – дополнительная информация о варианте
FORMAT – формат данных для каждого образца
cormapped14.bam – значения формата

Для анализа vcf файла воспользуемся bcftools stats:

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

a) Получилось 65482 варианта (number of records).
b) Из полученных вариантов 64463 являются однонуклеотидными заменами (number of SNPs).
c) Коротких вставок и делеций получилось 1019 (number of indels)

2. Фильтрация вариантов.

Отфильтруем полученные варианты:

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

`bcftools filter` – программа, которая отфильтрует варианты из входного файла по заданным параметрам
`-i'QUAL>30 && DP>50'` – фильтруем по качеству больше 30 и длине больше 50
`-o flt_variants.vcf` – вывод программы в указанный файл

Далее, чтобы проанализировать результат используем bcftools stats:

```
bcftools stats flter_variants.vcf > flter_variants.txt
```

Проанализируем файл `flter_variants.txt`:
a) Осталось вариантов – 1551 (2.38%)
b) Однонуклеотидных замен осталось – 1494 (2.32%)
c) Вставок и делеций осталось – 57 (5.6%)

3. Аннотация вариантов

Теперь с помощью Variant Effect Predictor (VEP) проаннотируем варианты:

Таблица 3. Summary statistics.

Category	Count
Variants processed	1551
Variants filtered out	0
Novel / existing variants	353 (22.8) / 1198 (77.2)
Overlapped genes	651
Overlapped transcripts	4897
Overlapped regulatory features	32

Всего в файле было найдено 1551 варианта (примерно в 42 раза меньше, чем с нефильтрованными данными).

Отфильтровано было 0.

Новые варианты – 353 (22.8%), а уже существующих – 1198 (77.2%).

Перекрывающихся генов – 651.

Перекрывающихся транскриптов – 4897 (их больше, чем генов, так как в один ген может входить несколько транскриптов).

Перекрывающихся регуляторных областей – 32.

Посмотрев на распределение эффектов мутаций (рис. 11). Видим, что мутации, как правило, возникают в интронах и некодирующих областях (34% и 11%). Также заметим, что у нас есть высокая доля кодирующих мутаций: синонимичные (10%) + миссенс (8%) + сдвиг рамки/возникновение стоп-кодона ~ 19%. Это указывает на селекцию в пользу экзонов.

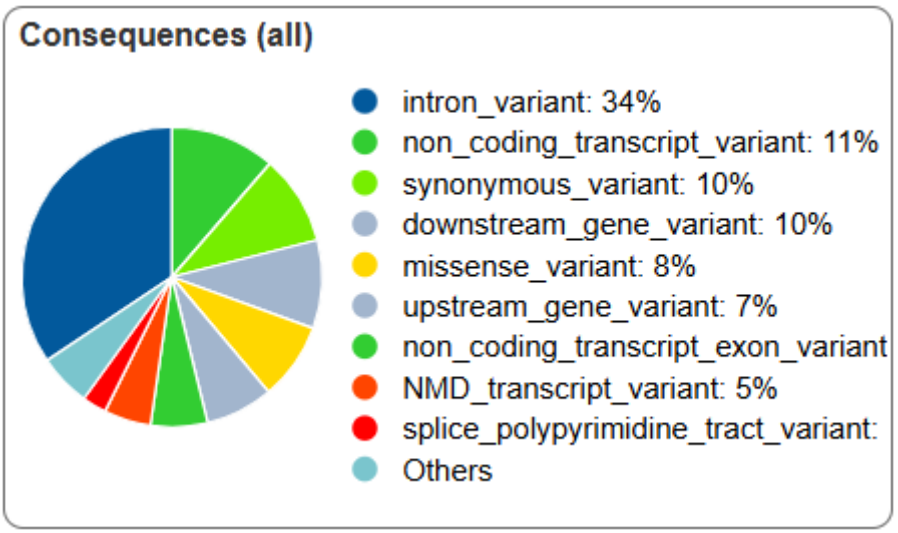

Рис. 11. Consequences (all).

Теперь рассмотрим эффекты мутаций в кодирующих областях (рис. 12). Большинство вариантов - это синонимичные мутации (53%) (не приведут к изменению белка) или миссенс мутации (45%) (могут вызвать изменение белка и вызвать мутационную изменчивость организма). При этом мы видим, что серьёзные мутации подавлены, на них приходится всего 2 процента.

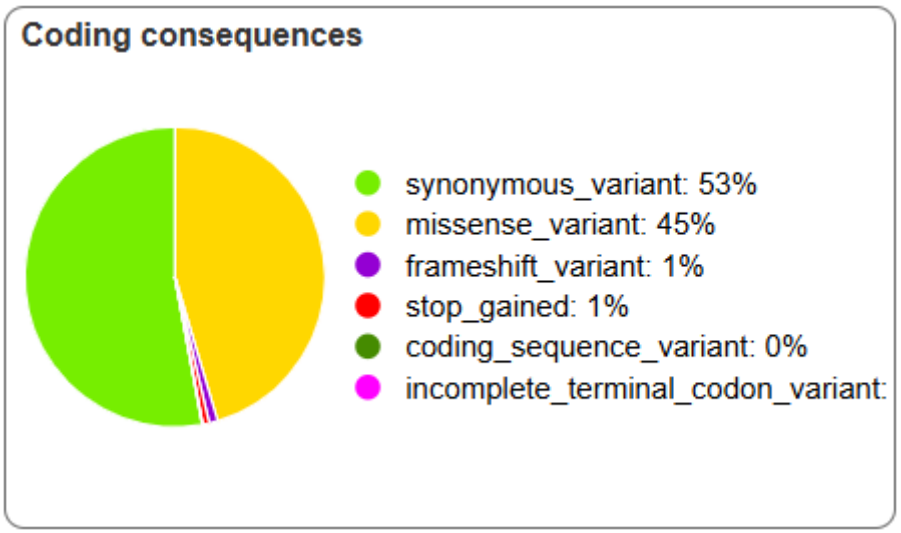

Рис. 12. Coding consequences.

Установили фильтр. Вариантов с IMPACT HIGH нашлось: 56. При этом стоит обратить внимание, что все мутации, за исключением одного, вызывают сдвиг рамки считывания или одиночную мутацию, образующую стоп-кодон. Чаще всего мы видим, что такие “опасные” замены встречаются в следующих генах: HSP90AA1, SYNE2, RPL36AL, RALGAPA1, ARHGAP5, TOX4. Как итог, мутации в этих генах повышают риск рака, так как функции этих белков связаны с клеточным циклом и нарушение их работы приведёт к образованию опухолей.

Практикум 13.

Анализ транскриптомов.

Задача практикума: построить экспрессионный профиль на основании данных секвенирования РНК.

1. Описание образца.

Таблица . Описание образца с помощью базы ENCODE.

ID образца РНК-чтений	ENCFF199IWW
Ссылка на информацию об образце	ENCSR795WFC
Организм и ткань (если есть)	*Mus musculus* strain B6CASTF1/J heart tissue male adult (18-20 months)
Стратегия секвенирования	RNA-seq (total RNA-seq)
Тип чтения	single-ended 100nt (одноконцевые чтения)
Цепь-специфичность	Strand-specific (reverse) (обратная цепь)

2. Проверка качества исходных чтений.

Задания этого практикума будут выполняться в директории `/mnt/scratch/NGS/paull/pr13`. Проанализируем качество исходных чтений:

```
fastqc ENCFF199IWW.fastq.gz
```

Получили файл , откроем его и проанализируем результат:

a) Всего чтений: 54017825

b) Посмотрев на полученный график мы можем заметить странность в начале. Межквартильное расстояние там минимальное, боксплоты отсутствуют (сильный разброс значений) и качество этих чтений на уровне 35, а дальше идут чтения с качеством ближе к 40 и межквартильное расстояние побольше и различимо глазом. Я думаю, такая странность может быть вызвана фазовым сдвигом (отставание некоторых молекул в синтезе) или наложением соседних сигналов друг на друга. Начиная с 72 нуклеотида и по 98 (последний нуклеотид) качество ухудшается. Усы боксплотов с 72 по 97 заходят на жёлтое поля, а усы 98 боксплота заходят в красную зону.

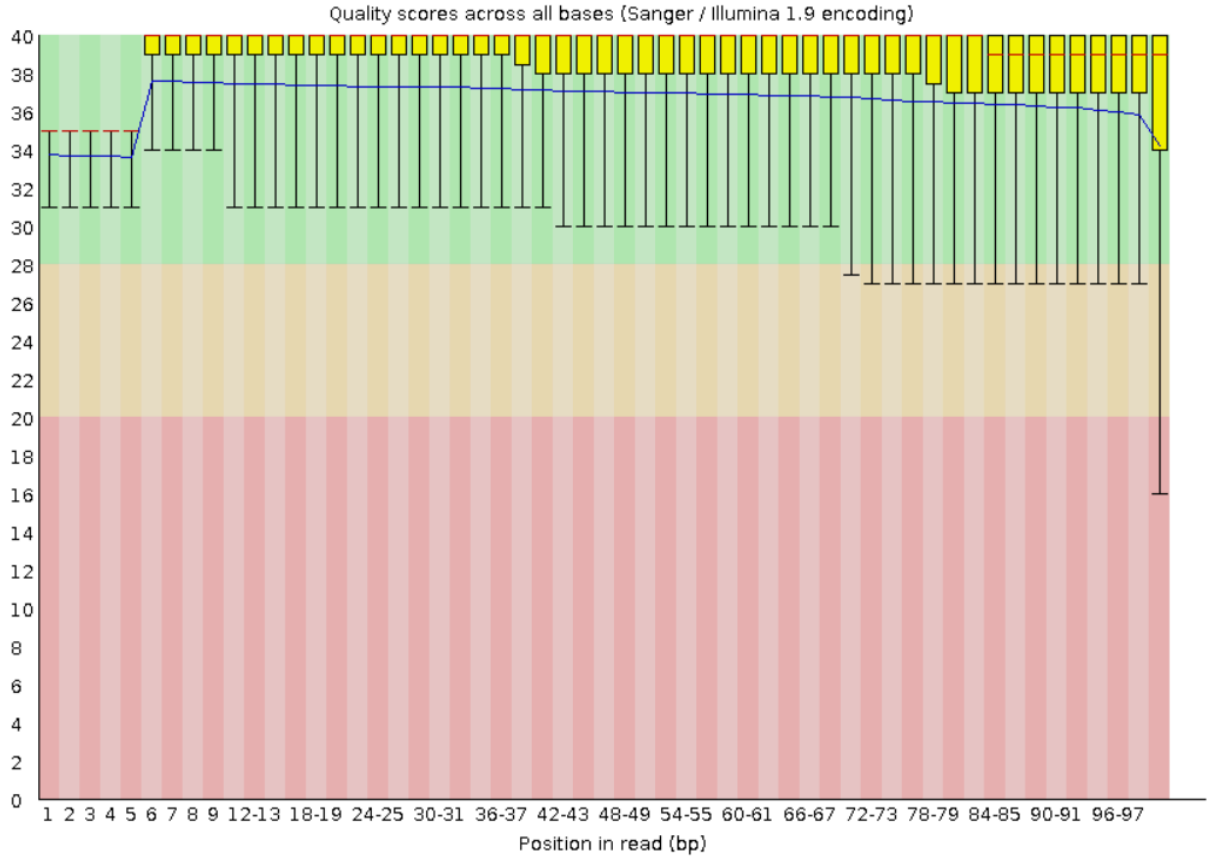

Рис. 13. Per base sequence quality.

c) Длина чтений равна 100 нуклеотидов (рис. 14).

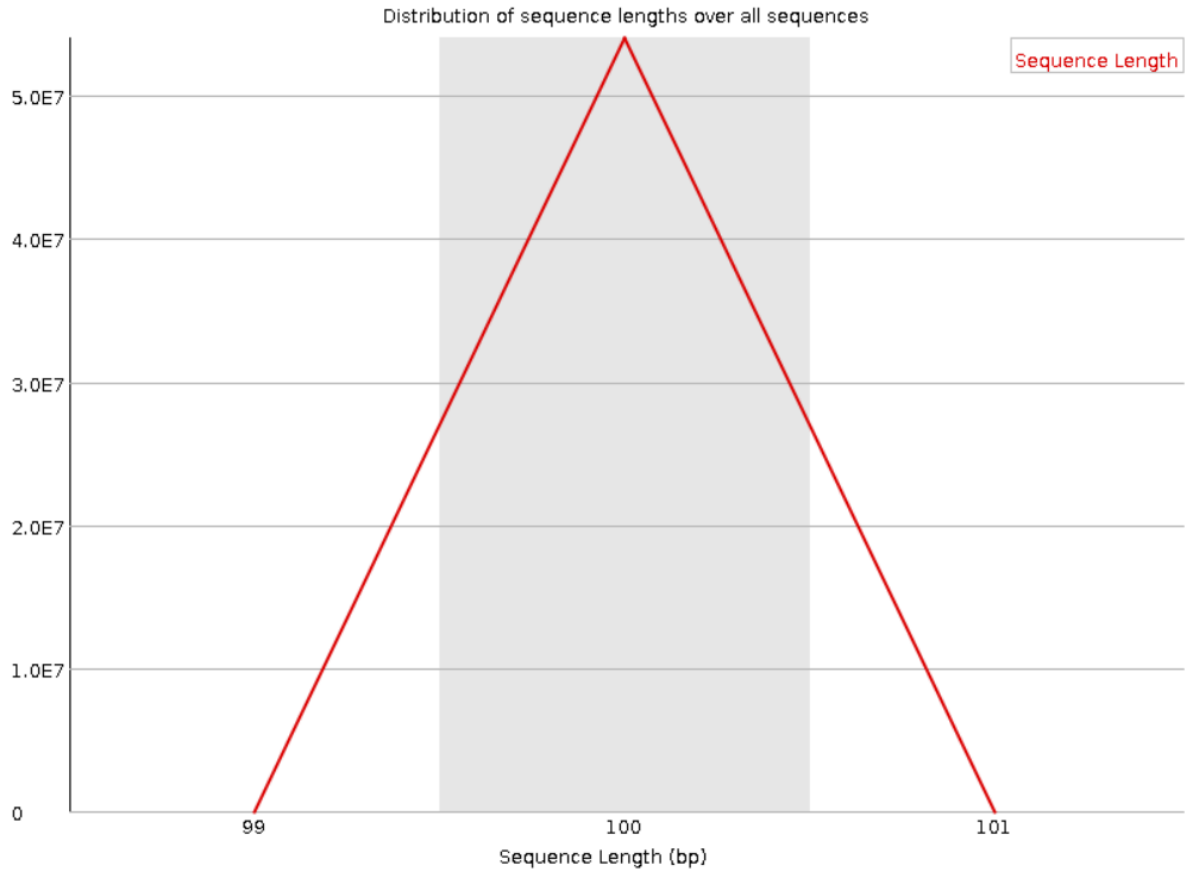

Рис. 14. Sequence Length Distribution.

3. Картирование чтений на референс.

Картирование чтений буду делать в директории /mnt/scratch/NGS/paull/rna_cart. Картируем программой hisat2:

```
hisat2 -x /mnt/scratch/NGS/paull/ref_genome/chr14_indexed -k 3 -U
../pr13/ENCFF199IWW.fastq.gz -S rna.sam 2> rna.log
```

`-x` – указывает префикс имён файлов с индексацией референса

`-k` - максимально возможное количество выравниваний, счет (score) которых больше или равен счету любого другого выравнивания

`-U` – указывает файл с чтениями

`-S` – выход в sam файл

`2>` – выводить логи в отдельный файл

Посмотрим, содержимое лог-файла:

```
54017825 reads; of these:
54017825 (100.00%) were unpaired; of these:
53088352 (98.28%) aligned 0 times
557837 (1.03%) aligned exactly 1 time
371636 (0.69%) aligned >1 times
1.72% overall alignment rate
```

Посмотрев в файл с логами, можно увидеть, что процент общего картирования очень маленький: 1.03% + 0.69% = 1.72%. Это может быть связано с тем, что у человека и у мыши (Mus musculus) мало совпадающих частей.

Дальше, переводим sam файл в bam: `samtools sort -o rna.bam rna.sam`

После этого проиндексируем bam файл: `samtools index rna.bam`

Дальше отберем только те чтения, которые картировались на хромосому:

```
samtools view -h -bS rna.bam 14 > rna14.bam
```

(Все параметры функции были указаны ранее в 5 пункте 2 части 11 практикума).

Чтобы посмотреть сколько чтений закартировалось на хромосому воспользуемся командой:

```
samtools flagstat rna14.bam > rna14.txt
```

Посмотрим содержимое полученного файла:

```
1314625 + 0 in total (QC-passed reads + QC-failed reads)
929473 + 0 primary
385152 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
1314625 + 0 mapped (100.00% : N/A)
929473 + 0 primary mapped (100.00% : N/A)
0 + 0 paired in sequencing
0 + 0 read1
0 + 0 read2
0 + 0 properly paired (N/A : N/A)
0 + 0 with itself and mate mapped
0 + 0 singletons (N/A : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

В файле rna14.txt можно увидеть, что всего картиовано было 1314625 чтений из 54402977 изначальных, что составляет 2.42% от всех чтений.

4. Поиск экспрессирующихся генов.

Теперь будем работать в директории /mnt/scratch/NGS/paull/Expression. Сначала копируем файл `Homo_sapiens.GRCh38.110.chr.gtf` к себе в директорию.

Теперь посмотрим на его содержимое. Сначала идет шапка:

`#!genome-build GRCh38.p14` - версия разметки

`#!genome-version GRCh38` - версия генома

`#!genome-date 2013-12` - дата публикации

`#!genome-build-accession GCA_000001405.29` - АС разметки

`#!genebuild-last-updated 2023-03` - последняя дата обновления

После шапки идет тело файла. В каждой строке находится информация о разметке, которая поделена на 9 столбцов:

seqname - название последовательности, где аннотирован ген

source - источник аннотации (программа)

feature - особенности гена

start - начало гена

end - конец гена

score – значение качества (оценка)

strand - цепь, на которой этот ген находится

frame – рамка считывания (с какого нуклеотида начинается)

attribute - дополнительная информация

Далее для того, чтобы посчитать количество картированных чтений на ген разметки сначала проиндексируем полученный bam файл: `samtools index rna14.bam`.

Далее воспользуемся программой htseq-count:

```
htseq-count -f bam -s reverse -m union -t gene rna14.bam Homo_sapiens.GRCh38.110.chr.gtf 1> counts.txt 2> rna1.log
```

`htseq-count` – выдает таблицу с тем, сколько чтений попало на какой ген

`-f bam` – входной файл в формате bam

`-s reverse` – была указана обратная цепь специфичность, поэтому параметр reverse

`-t gene` - определяет тип особенностей, которые будут браться (gene, так как нам надо посмотреть количество чтений картированных на гены)

`../Homo_sapiens.GRCh38.110.chr.gtf` – файл с разметкой

`1> counts.txt` – вывод отчета о работе программы

`2> rna1.log` – вывод ошибок

Получили два файла: текстовый и лог-файл. Рассмотрим последние строчки файла `counts.txt`:

```
__no_feature    1791
__ambiguous     83285
__too_low_aQual 0
```

```
__not_aligned   0
__alignment_not_unique  371636
```

Значения строк:

`no_feature` – чтение не легло ни на один элемент, который мы выбрали параметром -t.

ambiguous – двусмысленные чтения, которые одновременно перекрываются с несколькими аннотированными объектами.

`too_low_aQual` - отклонённые чтения из-за слишком низкого качества выравнивания.

`not_aligned` - чтения, которые не удалось выровнять с референсом.

`alignment_not_unique` – чтения, которые выровнялись несколько раз.

Теперь посчитаем количество чтений, которые попали на гены. Для этого из общего числа чтений вычтем `__no_feature, __ambiguous и __alignment_not_unique`: 1314625 - 1791 - 83285 - 371636 = 857913 (65.26%). Получаем 857913 чтения, то есть на гены попало примерно 65.26%.

Количество чтений, которые попали мимо генов – 1791 (не знаю, насколько это правильно, но я решил, что надо брать только значения из no_feature, так как именно они не попали на гены, у остальных уже другие проблемы).

5. Аннотация высоко экспрессируемых генов.

Получим в явном виде топ 10 самых высоко экспрессируемых генов:

```
head -n -5 counts.txt | sort -nk 2 -r > sorted_counts.txt
```

`head -n -5` - отбираем все строки кроме последних пяти

`sort -nk` – сортруем по убыванию, k параметр позволяет сортировать не по первой цифре, а по всему числу

Топ 10 самых высоко экспрессируемых генов:

```
ENSG00000276168 375710
ENSG00000197616 43142
ENSG00000092054 17639
ENSG00000200312 14801
ENSG00000224861 4005
ENSG00000165410 1358
ENSG00000092148 889
ENSG00000100852 836
ENSG00000090060 590
ENSG00000206588 449
```

Из них выберем второй ген ENSG00000197616 (43142), первый бел не белок-кодирующий. Этот ген кодирует тяжёлую цепь миозина сердечной мышца альфа-изоформы (Myosin Heavy Chain 6) (при этом важно отметить, что на третьем месте ещё одна изоформа тяжёлой цепи миозина, что логично, так как образец брался из сердечно-мышечной ткани мыши). Его основная функция - контроль сердечной деятельности: он преобразует химическую энергию молекул АТФ в механическую энергию сокращения сердца, может регулировать скорость и эффективность сердечных сокращений.

Далее визуализируем его в Genome Browser:

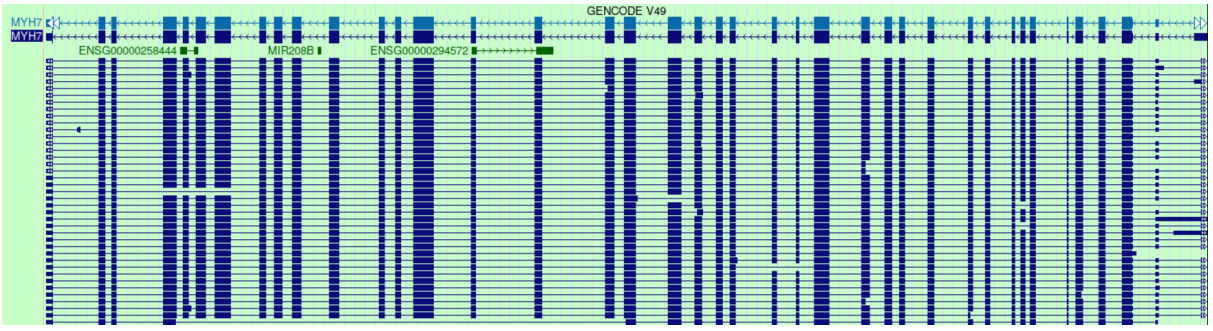

Рис. 15. Экзон-интронные структуры известных для гена ENSG00000197616.

Также рассмотрим трек консервативности:

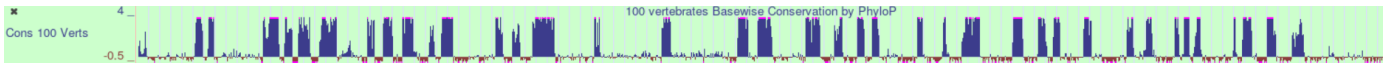

Рисунок 16. Трек консервативности ENSG00000197616.

По найденному треку консервативности видим, что ген наиболее консервативен в местах расположения экзонов (что логично, потому что необходимо максимально сохранить эти области).

Скрипт для практикумов 11-12.

```
#!/bin/bash
# Ussage ./script.sh ID chrN
```

Script

```
# for prepared corresponding files in current folder
hisat2-build chr${chrN}.fa chr${chrN} #индексация референса hisat2
samtools faidx chr${chrN}.fa #индексация референса с помощью samtools
fastqc ${ID}_1.fastq.gz ${ID}_2.fastq.gz #получаем файл для анализа качества изначальных чтений
TrimmomaticPE -phred${phred} ${ID}_1.fastq.gz ${ID}_2.fastq.gz ${ID}_1_paired.fastq.gz ${ID}_1_unpaired.fastq.gz ${ID}_2_paired.fastq.gz ${ID}_2_unpaired.fastq.gz TRAILING:${tr_trail} MINLEN:${tr_minlen} #триммируем парноконцевые чтения, удаляя с конца чтений с качеством ниже {tr_trailing} и удаляем чтения с длиной меньше {tr_minlen}
fastqc ${ID}_1_paired.fastq.gz ${ID}_1_unpaired.fastq.gz ${ID}_2_paired.fastq.gz ${ID}_2_unpaired.fastq.gz #получение файлов для анализа после триммирования
hisat2 -x chr${chrN} -1 ${ID}_1_paired.fastq.gz -2 ${ID}_2_paired.fastq.gz -p ${num_core} --${allign_type} > chr${chrN}_hst.sam #картирование чтений на хромосому {chrN} без учета сплайсинга
samtools sort -o chr${chrN}_hst.bam chr${chrN}_hst.sam #конвертируем файл sam в bam
samtools index chr${chrN}_hst.bam #индексируем файл bam
samtools flagstat chr${chrN}_hst.bam > analyse_chr${chrN}_bam.txt #создаем файл для анализа
samtools view -h -b chr${chrN}_hst.bam ${chrN} > chr${chrN}_hst_flt.bam #получаем чтения, которые картировались на референс
samtools view -f 0x2 -bS chr${chrN}_hst_flt.bam > chr${chrN}_hst_prop.bam #получаем только правильно картированные парные чтения
samtools flagstat chr${chrN}_hst_prop.bam > analyse_prop_bam.txt #получаем файл для анализа
samtools index chr${chrN}_hst_prop.bam #индексируем файл
bcftools mpileup -f Homo_sapiens.GRCh38.dna.chromosome.${chrN}.fa chr${chrN}_hst_prop.bam | bcftools call -mv -o file.vcf #ищем варианты и получаем файл vcf
bcftools stats file.vcf > file_stat.txt #получаем файл для анализа
bcftools filter -i'QUAL>${flt_qual} && DP>${flt_dp}' file.vcf > flt_variants.vcf #фильтрация вариантов по качестве больше {flt_qual}и глубиной больше {flt_dp}
bcftools stats flt_variants.vcf > variants_stat.txt #получаем файл для анализа
```

Options

```
ID=$1 #Id
chrN=$2 #номер хромосомы
phred=33 #quality score для триммирования
tr_trail=20 #обрезаем чтения с качеством ниже 20 с конца при триммировании
tr_minlen=50 #убираем чтения с длиной меньше 50 при триммировании
num_core=8 #количество ядер процессора
allign_type=no-spliced-alignment #запрещает большие разрывы внутри чтения при картировании
flt_qual=30 #фильтр качества для bsftools filter
flt_dp=50 # фильтр глубины для bsftools filter
```

Скрипт без комметариев после ‘#’ доступен по пути на kodomo: /mnt/scratch/NGS/paull/script/script.sh