Отчёт

ПРАКТИКУМ 11

1. Подготовка референса

1.1. Получение референса (индексация для hisat2):

Для картирования последовательности необходимо сначала провести индексацию генома. В моем случае в качестве референсного образца используется 4-я хромосома человека (Моё любимое число – 13. Где 13-я хромосома???)

hisat2-build Homo_sapiens.GRCh38.dna.chromosome.4.fa chr4

Команда: *hisat2-build* – индексирует референсный геном
Опции: *Homo_sapiens.GRCh38.dna.chromosome.4.fa* – файл с референсным геномом
chr4 – префикс, с которого начинаются названия индексных файлов
Выдача: *chr4.1.ht2 chr4.2.ht2 chr4.3.ht2 chr4.4.ht2 chr4.5.ht2 chr4.6.ht2 chr4.7.ht2 chr4.8.ht2* – индексные файлы

1.2. Индексация samtools:

Некоторые программы требуют предварительной индексации больших файлов в соответствии с их алгоритмами перед началом работы. Одной из таких программ является samtools. О какая привередливая! Поэтому нужна такая команда:

samtools faidx Homo_sapiens.GRCh38.dna.chromosome.4.fa

Получили файл *Homo_sapiens.GRCh38.dna.chromosome.4.fa.fai*, он содержит следующие 5 чисел разделенных табуляцией:

“4 190214555 56 60 61”

4 – номер хромосомы (точное имя хромосомы)
190214555 – длина хромосомы в нуклеотидах
56 – номер байта, с которого начинается нуклеотидная последовательность (перед ней начиная с нулевого байта идёт всё что расположено за символом “>”, например, название хромосомы (шапка))
60 – это число нуклеотидов в одной строке входного fasta файла
61 – длина строки в байтах (это значение на 1 больше, чем количество нуклеотидов, поскольку учитывается также символ переноса строки)

2. Чтения ДНК

2.1. Описание образца:

В базе NCBI в разделе SRA (https://www.ncbi.nlm.nih.gov/sra/) нашел свой образец по SRR ID, указанному в ведомости.

a) SRR ID образца ДНК-чтений: SRR10720404
b) Ссылка на информацию об образце из NCBI:
https://www.ncbi.nlm.nih.gov/sra/?term=SRR10720404
c) Прибор для секвенирования: Illumina Genome Analyzer IIx
d) Организм: *Homo sapiens (Традиционный сапиенс)*
e) Стратегия секвенирования: whole-exome sequencing (полноэкзомное)
f) Тип чтений: парно-концевые
g) Ожидаемое число чтений (spots): 38,518,929

2.2. Проверка качества исходных чтений:

Надо проанализировать 2 файла, так как у меня парно-концевые чтения. SRR10720404_1.fastq.gz (прямые чтения) и SRR10720404_2.fastq.gz (обратные чтения).

Запустил следующие команды:
```
fastqc SRR10720404_1.fastq.gz
fastqc SRR10720404_2.fastq.gz
```

Получил 2 .html файла, положил их в папку ~/public_html/term3 для быстрого доступа на зачете.

a) Кол-во пар чтений: 38518929
b) Одинаковое число в строке “Total Sequences” и для прямых и для обратных чтений (число чтений совпадает).
c) Качество пар чтений хорошее. В основном всё в зелёной области, только в конце немного в жёлтой (но это норма, что на концах чтения хуже), а в красной вообще ничего нет. Прямые и обратные чтения примерно одинаковы по качеству, хотя есть и различия. Программа FastQC отметила оба чтения галочками, значит чтения качественные. Медиана у большинства больше 36, при том у всех не меньше 34, а нижний квартиль у всех не меньше 32.

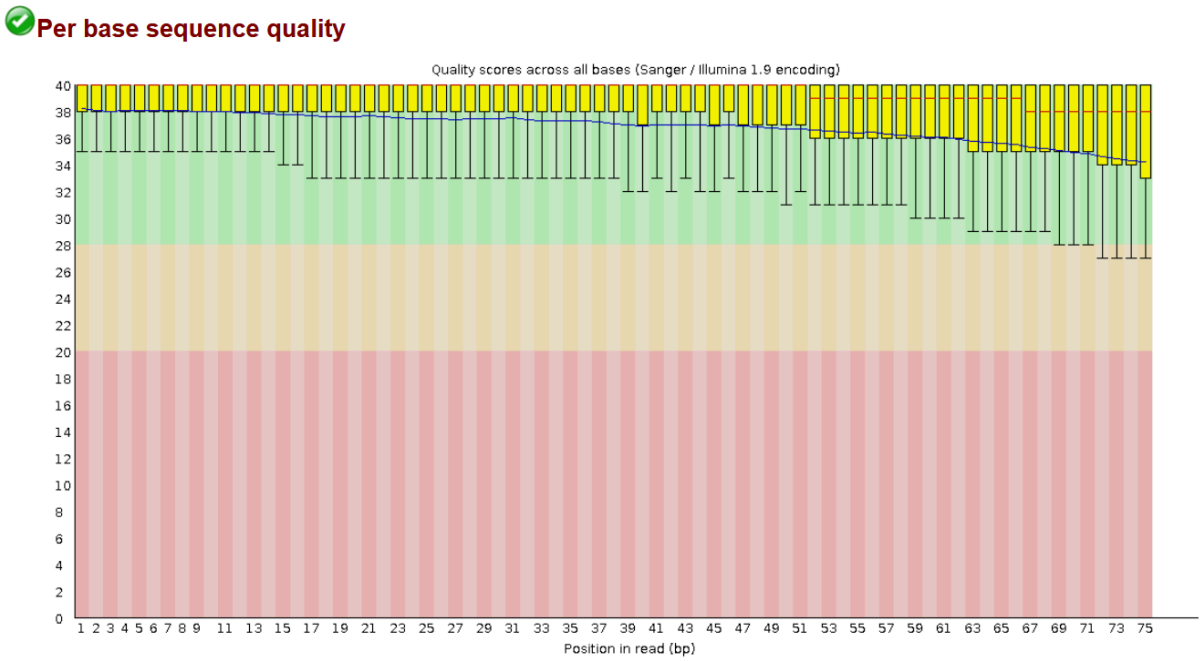

Рис 1. Прямые чтения

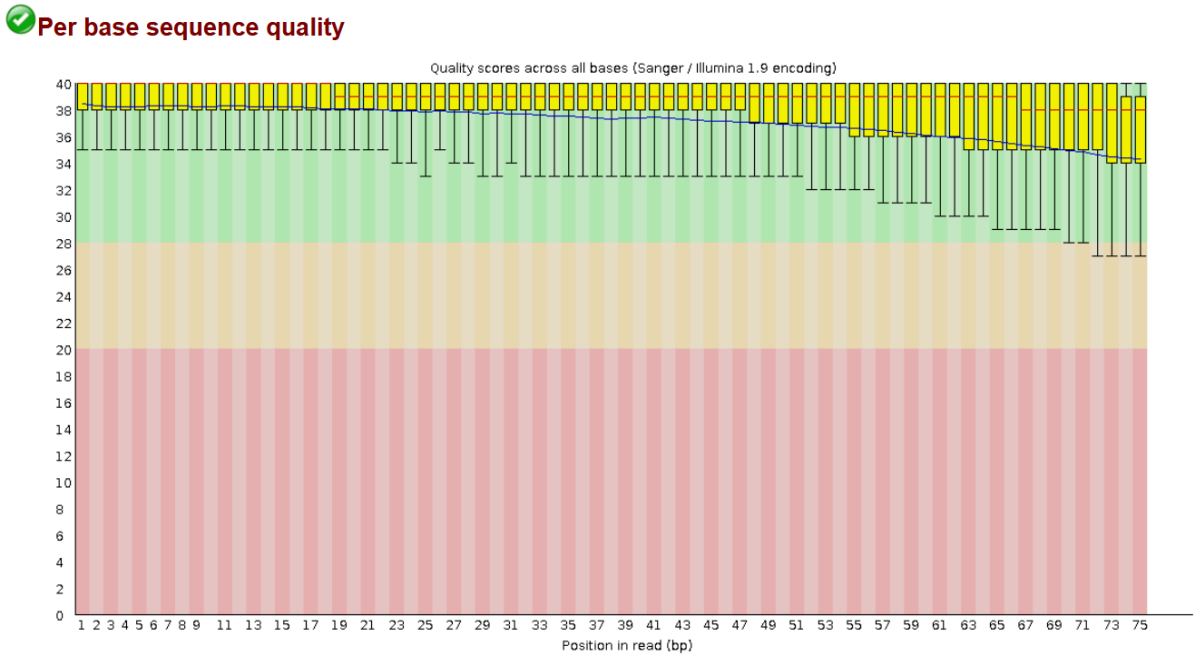

Рис 2. Обратные чтения

d) Распределение длины прочтений совпадает для прямых и обратных, поэтому привожу один график. Длина всех чтений = 75.

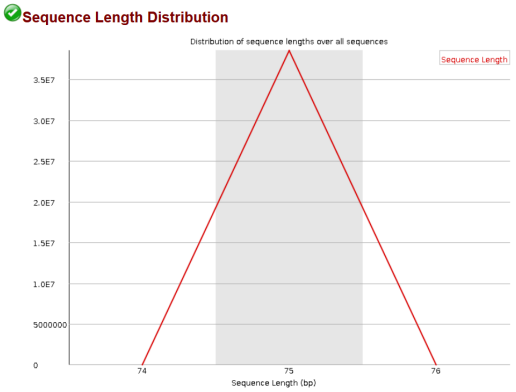

Рис 3. Распределение длин чтений

3. Фильтрация чтений

Проведу фильтрацию чтений с помощью trimmomatic, дабы попрактиковаться в использовании этого инструмента. Вообще, это нужно чтобы повысить как бы среднее качество чтений, то есть убрать плохие, не подходящие для дальнейшей работы. Хотя у меня чтения и так неплохие.

Удалил с конца чтений нуклеотиды с качеством ниже 20 и оставил чтения не меньше 50 нуклеотидов длиной.

TrimmomaticPE -phred33 SRR10720404_1.fastq.gz SRR10720404_2.fastq.gz tr_SRR10720404_1_pair.fastq.gz tr_SRR10720404_1_unpair.fastq.gz tr_SRR10720404_2_pair.fastq.gz tr_SRR10720404_2_ unpair.fastq.gz TRAILING:20 MINLEN:50

Команда: *TrimmomaticPE* – Trimmomatic для парноконцевых чтений
Опции: *-phred33* – quality score (оценка качества)
TRAILING:20 – удаление нуклеотидов с конца с качеством ниже 20
MINLEN:50 – удаление чтений с длиной меньше 50
Выдача: *tr_SRR10720404_1_pair.fastq.gz tr_SRR10720404_1_unpair.fastq.gz tr_SRR10720404_2_pair.fastq.gz tr_SRR10720404_2_ unpair.fastq.gz* – отфильтрованные файлы
Input Read Pairs: 38518929 Both Surviving: 37136668 (96.41%) Forward Only Surviving: 596429 (1.55%) Reverse Only Surviving: 377575 (0.98%) Dropped: 408257 (1.06%) – выдача в командной строке после триммирования

После триммирования одно из двух чтений может быть отброшено (не отвечая длине 50 после обработки), в результате чего оставшееся чтение останется без пары. Программа сохранит такие чтения в отдельный файл, который будет обозначен как unpair.

4. Проверка качества триммированных чтений

Чтобы проверить качество после триммирования я запустил следующую команду

*fastqc tr**

a) Какое количество пар чтений осталось (paired) в штуках: 37136668
b) Какой процент пар чтений остался (paired) (процент от исходного количества пар чтений): 96,41%
c) Парные чтения после триммирования выглядят очень хорошо, всё качество в зелёной области, в то время как непарные выглядят плохо – есть захождения в красную область.
d) Если сравнить парные чтения до и после триммирования, то видно, что качество стало лучше. Больше нет заходов в желтую область в конце.

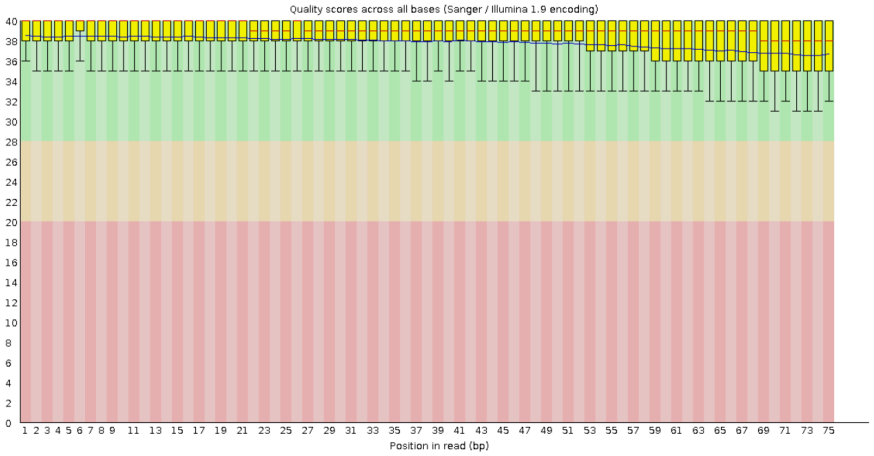

Рис 6. Прямые парные чтения после триммирования

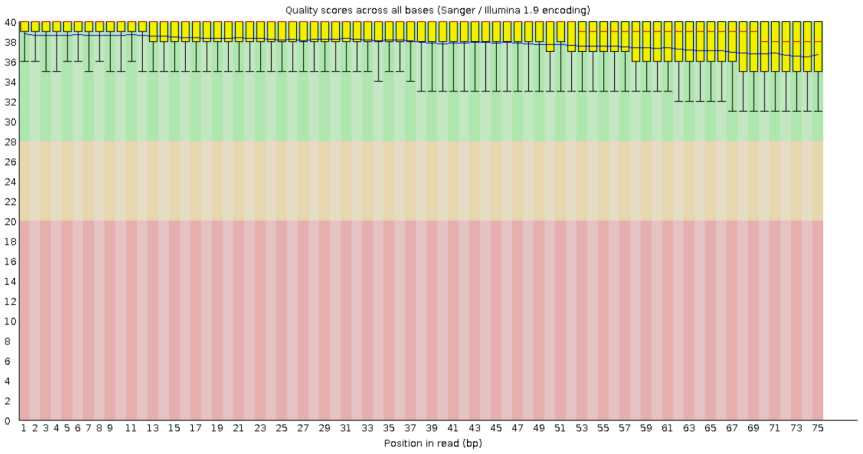

Рис 7. Обратные парные чтения после триммирования

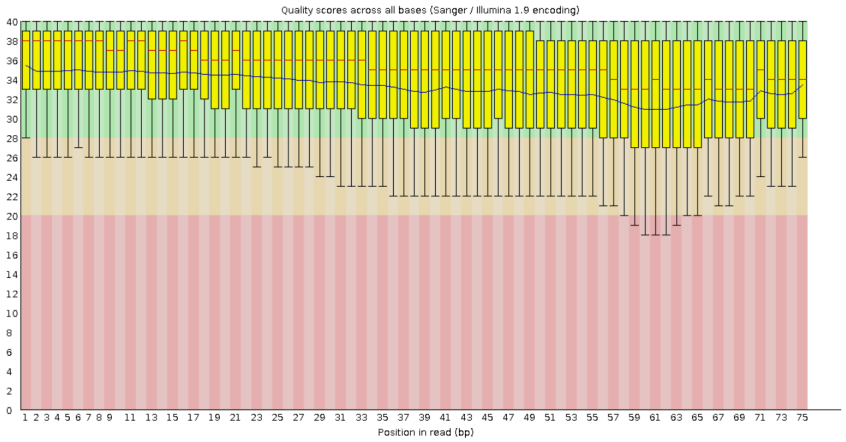

Рис 8. Прямые непарные чтения после триммирования

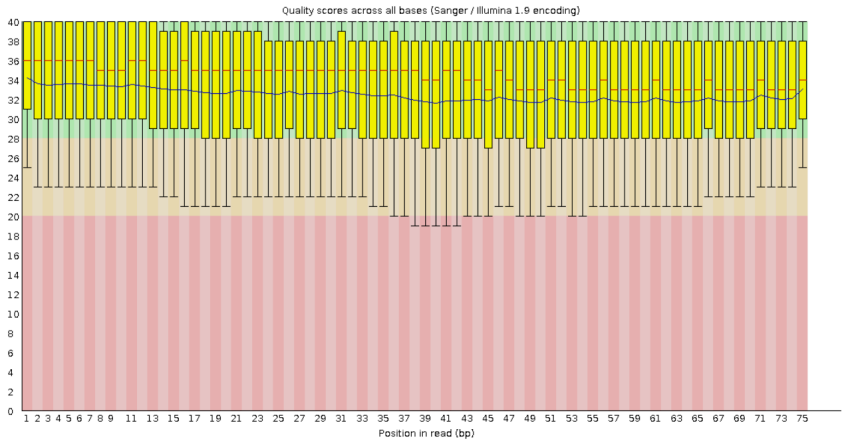

Рис 9. Обратные непарные чтения после триммирования

е) Почти все парные чтения остались длиной 75 (хотя видно небольшие пик на 70, например), а вот непарных много других длин (много разных пиков). Но большинство всё равно длиной 75. Привожу графики только для прямых чтений, так как у обратных картина очень похожа.

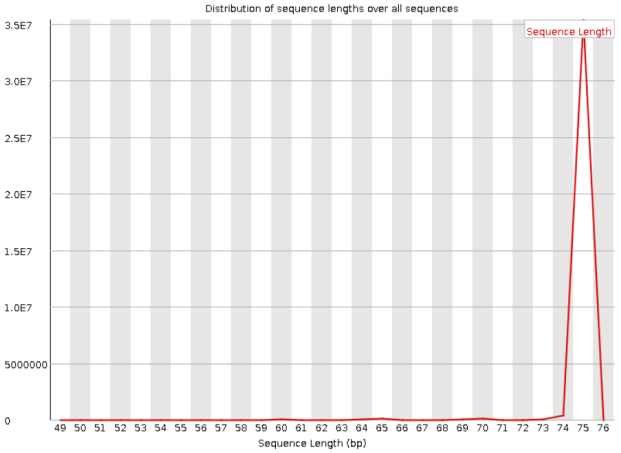

Рис 4. Распределение длин прямых парных чтений после триммирования

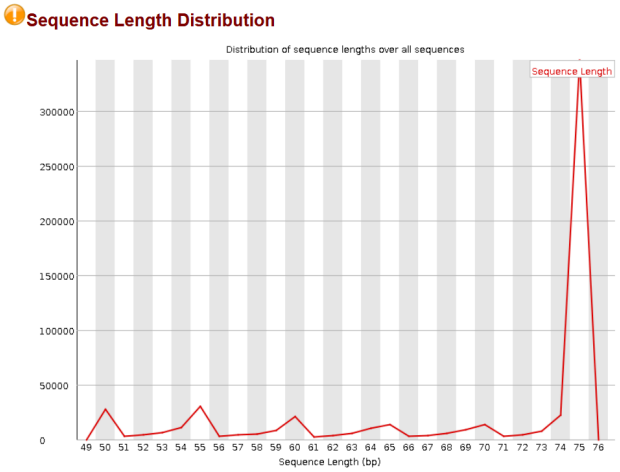

Рис 5. Распределение длин прямых непарных чтений после триммирования

ПРАКТИКУМ 12

1. Картирование чтений на референсный геном

Я картировал триммированные парные чтения на референсный геном с помощью следующей команды:

hisat2 -x chr4 -1 tr_SRR10720404_1_pair.fastq.gz -2 tr_SRR10720404_2_pair.fastq.gz -p 4 --no-spliced-alignment > chr4_map.sam 2> logs.txt

Команда: *hisat2* – выравнивает чтений на референсный геном
Опции: *-x chr4* – указывает на префикс, с которого начинаются индексные файлы.
-1 tr_SRR10720404_1_pair.fastq.gz – указывает на файл с прямыми парными чтениями после триммирования.
-2 tr_SRR10720404_2_pair.fastq.gz – указывает на файл с обратными парными чтениями после триммирования.
-p 4 – задает количество потоков (в данном случае 4), которые будут использоваться для параллельной обработки, что может ускорить процесс выравнивания.
--no-spliced-alignment – указывает программе не выполнять выравнивание с учетом сплайсинга, то есть не учитывать экзоны и интроны в последовательностях.

> chr4_map.sam – перенаправляет стандартный вывод (результаты выравнивания) в файл “chr4_map.sam”, который будет содержать выровненные последовательности в формате SAM.
2> logs.txt – перенаправляет стандартный поток ошибок (ошибки и предупреждения) в файл “logs.txt”
Выдача: *chr4_map.sam* – файл с парными чтениями, картированными на референсный геном (на 4-ю хромосому).
logs.txt – файл с стандартным потоком ошибок, кратким review работы программы.

2. Конвертация sam в bam

2.1. Описание sam/bam файла

a) Файл SAM весит около 15,48 ГБ

Применил следующую команду для конвертации sam в bam:

samtools sort -o chr4_map.bam chr4_map.sam

Команда: *samtools sort* – сортировка (конвертирование) файла
Опции: *-o* – запись итога работы программы в файл
chr4_map.bam – файл, в который запишется итог
chr4_map.sam – файл, который подаётся на вход программе
Выдача: *chr4_map.bam.bai* – получившийся файл

b) Файл BAM весит около 4,24 ГБ (весит меньше, так как BAM это сжатое бинарное представление SAM)

2.2. Индексирование bam файла

С помощью следующей команды я проиндексировал полученный bam файл:

samtools index chr4_map.bam

Команда: *samtools index* – индексирование файла
Выдача: *chr4_map.bam.bai* – получившийся файл

3. Анализ bam файла

BAM (Binary Alignment Map) — это сжатое бинарное представление SAM (Sequence Alignment Map). Чтобы его проанализировать, я использовал следующую команду для запуска программы samtools:

samtools flagstat chr4_map.bam > chr4_map_flagstat.txt

Команда: *samtools flagstat* – индексирование файла
Выдача: *chr4_map_flagstat.txt* – получившийся файл

Вот что содержит получившийся файл:
75603497 + 0 in total (QC-passed reads + QC-failed reads)
74273336 + 0 primary
1330161 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
6143366 + 0 mapped (8.13% : N/A)
4813205 + 0 primary mapped (6.48% : N/A)
74273336 + 0 paired in sequencing
37136668 + 0 read1
37136668 + 0 read2
4150970 + 0 properly paired (5.59% : N/A)
4258270 + 0 with itself and mate mapped
554935 + 0 singletons (0.75% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)

1) Число в поле «in total»: общее количество чтений, которые прошли проверку, плюс количество тех, которые не прошли (всего обработано чтений, при парном выравнивании учитываются оба чтения).
2) Сколько чтений (не пар!) поступило на картирование: *75603497* (из первой строки файла)
3) Число чтений, картированых на референс в корректных парах в штуках: *4150970* (строка “properly paired”)
4) Число чтений, картированых на референс в корректных парах в процентах относительно поступивших на картирование: *5,59%*

4. Получение чтений, картированных на хромосому

Властью, данной мне, при помощи Всевышнего, а также следующей команды, я получил чтения, картированные только на данную мне хромосому:

samtools view -h -bS chr4_map.bam 4 > chr4_map_only.bam

Команда: *samtools view* – выдаёт все чтения из файла, картированные на референс
Опции: *-h* – вывод вместе с заголовком

-bS – -b (вывод в формате bam) + -S (автоматическое определение формата передаваемого файла)
chr4_map.sam – файл, который подаётся на вход программе
Выдача: *chr4_map_only.bam* – файл с картированными на 4 хромосому чтениями

Используя следующую команду чтобы проанализировать полученный файл с картированными на 4 хромосому чтениями:

samtools flagstat chr4_map_only.bam > chr4_map_only_flagstat.txt

Вот что содержит получившийся файл:
6698301 + 0 in total (QC-passed reads + QC-failed reads)
5368140 + 0 primary
1330161 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
6143366 + 0 mapped (91.72% : N/A)
4813205 + 0 primary mapped (89.66% : N/A)
5368140 + 0 paired in sequencing
2684070 + 0 read1
2684070 + 0 read2
4150970 + 0 properly paired (77.33% : N/A)
4258270 + 0 with itself and mate mapped
554935 + 0 singletons (10.34% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)

Произошли некие, так называемые изменения. Во первых, уменьшилось подаваемое на вход число чтений (теперь их *6698301*). Во-вторых, процент чтений, картированных на референс в корректных парах, от общего количества чтений, поступивших на картирование, сильно возрос (с *5,59%* до *77.33%*).

5. Получение только правильно картированных пар чтений

Запустил samtools, для уже картированных на референсный геном чтений, чтобы получить только правильно картированные пары чтений (то есть чтения из одной пары должны картироваться недалеко друг от друга и быть направлены друг к другу):

samtools view -f 2 -bS chr4_map_only.bam > chr4_map_true.bam

Команда: *samtools view* – выдаёт все чтения из файла, картированные на референс
Опции: *-f 2* – вывод только тех чтений, которые правильно выровнены с референсом (прошли по критерию, соответствуют PROPER_PAIR)
-bS – -b (вывод в формате bam) + -S (автоматическое определение формата передаваемого файла)
chr4_map_only.bam – файл, который подаётся на вход программе
Выдача: *chr4_map_true.bam* – файл только с правильно картированными на 4 хромосому чтениями

Взяв всю свою волю в кулак, я использую команду чтобы проанализировать полученный файл с правильно картированными на 4 хромосому чтениями:

samtools flagstat chr4_map_true.bam > chr4_map_true_flagstat.txt

Вот что содержит получившийся файл:
5134416 + 0 in total (QC-passed reads + QC-failed reads)
4150970 + 0 primary
983446 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
5134416 + 0 mapped (100.00% : N/A)
4150970 + 0 primary mapped (100.00% : N/A)
4150970 + 0 paired in sequencing
2075485 + 0 read1
2075485 + 0 read2
4150970 + 0 properly paired (100.00% : N/A)
4150970 + 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)

Снова изменилось количество чтений на входе. Теперь процент чтений, соответствующих референсу, от общего числа триммированных достиг 100%, и все процентные показатели достигли своих предельных значений (100 и 0). Это логично, так как мы старались наилучшим образом отфильтровать файлы, чтобы оставить только нужно, выбрали правильные картированные пары и теперь убедились, что это действительно так.

Далее, приложив огромные усилия, я проиндексировал полученный bam файл, содержащий только правильно картированные чтения на 4-ю хромосомы.

samtools index chr4_map_true.bam

Команда: *samtools index* – индексирование файла
Выдача: *chr4_map_true.bam.bai* – получившийся файл

6. Получение чтений, картированных только в границы экзома

Я использовал BAM-файл, полученный на предыдущем этапе (с чтениями, которые были картированы на референс только для моей хромосомы и в правильных парах). В этом файле мне нужно было оставить только те чтения, которые были сопоставлены в пределах экзома. Принеся жертву Одину, я выполнил следующую команду:

bedtools intersect -a chr4_map_true.bam -b seqcap_hg38.bed > bed_chr4_map_true.bam

Команда: *bedtools intersect* – кроватные тулы рулят

Набравшись сил, я использую команду чтобы проанализировать полученный файл:

samtools flagstat bed_chr4_map_true.bam > bed_chr4_map_true_flagstat.txt

Вот что содержит получившийся файл:

2221246 + 0 in total (QC-passed reads + QC-failed reads)
2200954 + 0 primary
20292 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
2221246 + 0 mapped (100.00% : N/A)
2200954 + 0 primary mapped (100.00% : N/A)
2200954 + 0 paired in sequencing
1099073 + 0 read1
1101881 + 0 read2
2200954 + 0 properly paired (100.00% : N/A)
2200954 + 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)

Снова изменилось число чтений на входе (теперь их *2221246*). Также отличие от предыдущих анализов, здесь наблюдается различие в количестве "read1" и "read2". Видимо некоторые чтения из пар были отфильтрованы, так как не удовлетворяли текущим требованиям (располагались вне границ экзома).

ПРАКТИКУМ 13

1. Получение вариантов

Я набил полные погреба терпением, запустил следующую команду и принялся резко нейтрально ждать:

bcftools mpileup -f Homo_sapiens.GRCh38.dna.chromosome.4.fa chr4_map_true.bam | bcftools call -mv -o chr4_map_true.vcf

Команда: *bcftools mpileup* – создает vcf файл с вероятностями
bcftools call – выбирает только указанные строки из выдачи предыдущей программы
Опции: *-f* – указание на референс
-mv – ищет и выводит мультиаллельные и редкие варианты
-o – output
Выдача: *chr4_map_true.vcf* – файл с вариантами

Откроем файл и опишем его начинку:
~~Карамель в сочетании с нежным муссом из фейхоа~~

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##bcftoolsVersion=1.16+htslib-1.16
##bcftoolsCommand=mpileup -f Homo_sapiens.GRCh38.dna.chromosome.4.fa chr4_map_true.bam
##reference=file://Homo_sapiens.GRCh38.dna.chromosome.4.fa
##contig=<ID=4,length=190214555>
##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=RPBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Read Position Bias (closer to 0 is better)">
##INFO=<ID=MQBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Mapping Quality Bias (closer to 0 is better)">
##INFO=<ID=BQBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Base Quality Bias (closer to 0 is better)">
##INFO=<ID=MQSBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Mapping Quality vs Strand Bias (closer to 0 is better)">
##INFO=<ID=NMBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Number of Mismatches within supporting reads (closer to 0 is better)">
##INFO=<ID=SCBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Soft-Clip Length Bias (closer to 0 is better)">
##INFO=<ID=FS,Number=1,Type=Float,Description="Phred-scaled p-value using Fisher's exact test to detect strand bias">
##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=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=1.16+htslib-1.16
##bcftools_callCommand=call -mv -o chr4_map_true.vcf; Date=Mon Dec 23 11:25:05 2024
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  chr4_map_true.bam
4       10165   .       G       C       225.417 .       DP=25;VDB=2.39594e-06;SGB=-0.683931;MQSBZ=-0.426401;FS=0;MQ0F=0.04;AC=2;AN=2;DP4=0,0,2,11;MQ=55 GT:P
L       1/1:255,39,0
4       11911   .       C       A       5.04598 .       DP=1;SGB=-0.379885;FS=0;MQ0F=0;AC=2;AN=2;DP4=0,0,1,0;MQ=60      GT:PL   1/1:33,3,0
4       15444   .       G       A       3.77034 .       DP=2;SGB=-0.379885;RPBZ=1;MQBZ=0;MQSBZ=0;BQBZ=1;NMBZ=0;SCBZ=0;FS=0;MQ0F=0;AC=1;AN=2;DP4=0,1,1,0;MQ=6
0       GT:PL   0/1:34,0,31
4       15445   .       A       G       3.77163 .       DP=2;SGB=-0.379885;RPBZ=1;MQBZ=0;MQSBZ=0;BQBZ=0;NMBZ=0;SCBZ=0;FS=0;MQ0F=0;AC=1;AN=2;DP4=0,1,1,0;MQ=6
0       GT:PL   0/1:34,0,34
4       15859   .       C       A       10.7923 .       DP=1;SGB=-0.379885;FS=0;MQ0F=0;AC=2;AN=2;DP4=0,0,0,1;MQ=60      GT:PL   1/1:40,3,0
4       16725   .       A       G       29.3999 .       DP=4;VDB=0.02;SGB=-0.453602;RPBZ=-1.63299;MQBZ=0;MQSBZ=0;BQBZ=1;NMBZ=0;SCBZ=0;FS=0;MQ0F=0;AC=1;AN=2;
DP4=0,2,2,0;MQ=60       GT:PL   0/1:62,0,56
:
```

Рис 6. Начинка vcf файла (верхняя часть)

1. Шапка - строчки начинаются с #, там расположена информация о данных
2. Строка с названием столбцов
3. Основная информация о вариантах (различные столбцы):
CHROM - номер хромосомы
POS - позиция варианта
ID - может быть любая информация о варианте, но обычно пустой (в основном стоят “.”)
REF - буква стоящая в референсе
ALT - альтернативный вариант аллеля
QUAL - качество
INFO - различные характеристики варианта
FORMAT - параметры варианта

Теперь пришёл черёд проанализировать этот файл более серьёзно. Пишу команду:

bcftools stats chr4_map_true.vcf > stataaa.txt

```
# This file was produced by bcftools stats (1.16+htslib-1.16) and can be plotted using plot-vcfstats.
# The command line was: bcftools stats  chr4_map_true.vcf
#
# Definition of sets:
# ID    [2]id   [3]tab-separated file names
ID      0       chr4_map_true.vcf
# SN, Summary numbers:
#   number of records   .. number of data rows in the VCF
#   number of no-ALTs   .. reference-only sites, ALT is either "." or identical to REF
#   number of SNPs      .. number of rows with a SNP
#   number of MNPs      .. number of rows with a MNP, such as CC>TT
#   number of indels    .. number of rows with an indel
#   number of others    .. number of rows with other type, for example a symbolic allele or
#                          a complex substitution, such as ACT>TCGA
#   number of multiallelic sites     .. number of rows with multiple alternate alleles
#   number of multiallelic SNP sites .. number of rows with multiple alternate alleles, all SNPs
#
#   Note that rows containing multiple types will be counted multiple times, in each
#   counter. For example, a row with a SNP and an indel increments both the SNP and
#   the indel counter.
#
# SN    [2]id   [3]key  [4]value
SN      0       number of samples:      1
SN      0       number of records:      106127
SN      0       number of no-ALTs:      0
SN      0       number of SNPs: 104609
SN      0       number of MNPs: 0
SN      0       number of indels:       1518
SN      0       number of others:       0
:
```

Рис 7. Начинка файла stats.txt (верхняя часть)

a) Число получившихся вариантов: *106127*
b) Число однонуклеотидных замен: *104609*
c) Число коротких вставок и делеций: *1518*

2. Фильтрация вариантов

Фильтрую варианты по качеству и покрытию при помощи этого заклинания:

bcftools filter -i 'QUAL>30 && DP>50' chr4_map_true.vcf -o variants.vcf

Команда: *bcftools filter* – команда для фильтрации

Опции: *-i 'QUAL>30 && DP>50'* – фильтрация по качеству не ниже 30 и глубиной покрытием не ниже 50
-o – output
Выдача: *variants.vcf* – файл с вариантами

Анализируем *variants.vcf* :

bcftools stats variants.vcf > variants.txt

```
# SN    [2]id   [3]key  [4]value
SN      0       number of samples:      1
SN      0       number of records:      2592
SN      0       number of no-ALTs:      0
SN      0       number of SNPs: 2490
SN      0       number of MNPs: 0
SN      0       number of indels:       102
SN      0       number of others:       0
SN      0       number of multiallelic sites:   7
SN      0       number of multiallelic SNP sites:       3
```

Рис 8. Часть файла variants.txt, содержащая необходимые для практикума значения

a) Сколько осталось вариантов после фильтрации: *2592* или 2,44%
b) Сколько осталось однонуклеотидных замен: *2490* или 2,38%
c) Сколько осталось коротких вставок и делеций: *102* или 6,72%

Довольно мало в процентах, хм)) Ну это так, на обывательский взгляд.

3. Аннотация вариантов

Провел аннотацию файла с вариантами при помощи Variant Effect Predictor (VEP).

Category	Count
Variants processed	2592
Variants filtered out	0
Novel / existing variants	516 (19.9) / 2076 (80.1)
Overlapped genes	796
Overlapped transcripts	4642
Overlapped regulatory features	19

Рис 9. Сводная статистика (1)\

В результате анализа, проведенного с помощью Variant Effect Predictor (VEP), было обработано 2592 варианта, все из которых были аннотированы, так как не было отфильтрованных данных. Из них 516 вариантов являются новыми и не имеют предыдущих описаний, в то время как 2076 вариантов уже были аннотированы ранее. Обнаружено 796 перекрывающихся генов и 4642 транскрипта. Также выявлено 19 регулярных областей.

Анализ показывает, что значительная часть вариантов (около 20%) является новыми, что может указывать на потенциально важные и интересные генетические изменения в образце, требующие дальнейшего изучения. Высокое количество перекрывающихся генов и транскриптов может свидетельствовать о сложной генетической структуре исследуемого материала. Наличие регулярных областей также может быть важным для понимания функциональной значимости выявленных вариантов.

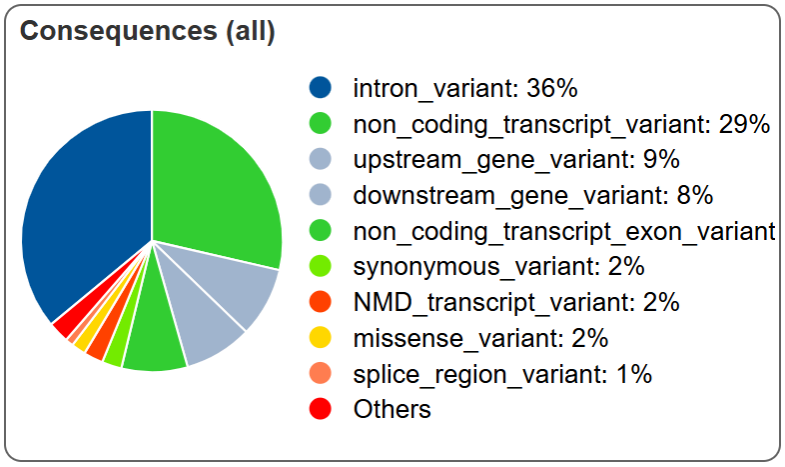

Рис 10. Сводная статистика (2) – распределение мутаций

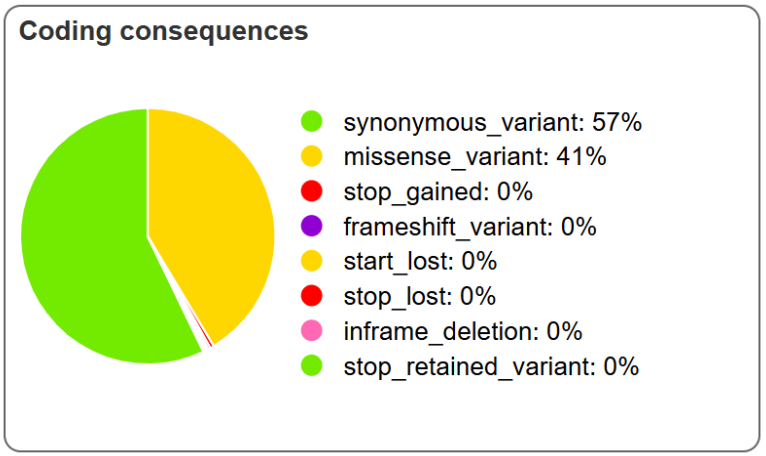

Рис 11. Сводная статистика (3) – распределение мутаций в кодирующих участках

Также можно отметить, что наиболее значительную долю (36%) составляют интронные варианты, что может указывать на важность регуляторных элементов в геномной структуре (характерно для высших эукариот). Эти изменения могут влиять на регуляцию экспрессии генов, сплайсинг и стабильность мРНК. Поскольку интроны часто содержат регуляторные элементы, их вариации могут иметь значительное влияние на функциональность генов.

Высокий процент некодирующих вариантов (29%) также подчеркивает значимость некодирующих РНК в генетической регуляции (включая микроРНК и длинные некодирующие РНК).

Upstream_gene_variant (9%) – эти изменения могут влиять на промоторные регионы и, следовательно, на уровень экспрессии гена. Они могут быть важны для понимания регуляции генов в ответ на различные сигналы.

Downstream_gene_variant (8%) – подобно upstream-вариантам, они могут влиять на регуляцию экспрессии, а также на стабильность мРНК и взаимодействие с другими молекулами.

Synonymous_variant (2%) – синонимичные замены. Хотя эти изменения не влияют на структуру белка, они могут оказывать влияние на стабильность мРНК и скорость трансляции, что может иметь косвенные эффекты на экспрессию.

Missense_variant (2%) – приводя к замене одной аминокислоты в белке на другую, эти изменения могут иметь значительное влияние на функцию белка, потенциально приводя к заболеваниям или изменению фенотипа.

Splice_region_variant (1%) – варианты, находящиеся в области сплайсинга. Эти изменения могут влиять на процесс сплайсинга, что может привести к образованию альтернативных изоформ белка или даже к потере функции.

В кодирующих участках наблюдается высокая доля синонимичных вариантов (57%) и относительно низкая доля миссенс-вариантов (41%). Это может свидетельствовать о том, что большинство изменений не приводят к значительным изменениям в белковой структуре, это характерно в целом для генетического кода, так что логично.

Полное отсутствие таких вариантов, как stop_gained, frameshift_variant и других, указывает на то, что в анализируемых последовательностях не было обнаружено серьезных мутаций, которые могли бы привести к потере функции белка. Высокий процент новых вариантов и разнообразие типов изменений подчеркивают необходимость дальнейшего изучения для понимания их

функциональной значимости и потенциального влияния на фенотипические проявления.

Однако, у меня нет вариантов с IMPACT HIGH, но есть с IMPACT MODERATE, например missense_variant: 2%.

ПРАКТИКУМ 15

1. Описание образца

Я нашел ID своего образца секвенирования РНК в базе ENCODE.

a. ID образца РНК-чтений: ENCSR795WFC
b. Ссылка на информацию об образце:
https://www.encodeproject.org/files/ENCFF641WPY/ (файл)
https://www.encodeproject.org/experiments/ENCSR795WFC/ (эксперимент)
c. Организм и ткань: Mus musculus strain B6CASTF1/J сердечная ткань взрослого МЫША – мужик возрастом 18-20 месяцев (он так стар, что экспериментаторы уже забыли сколько ему)
d. Стратегия секвенирования: тотальное секвенирование РНК
e. Одноконцевые чтения
f. Цепь-специфичность: Strand-specific (reverse) – есть специфичность (обратная цепь)

2. Проверка качества исходных чтений

Проанализировал качество исходных чтений с помощью программы fastqc. Использовал такую вот удивительную команду, которой меня научили в Хогвартсе:

```
fastqc ENCFF641WPY.fastq.gz
```

Число чтений: 48692928

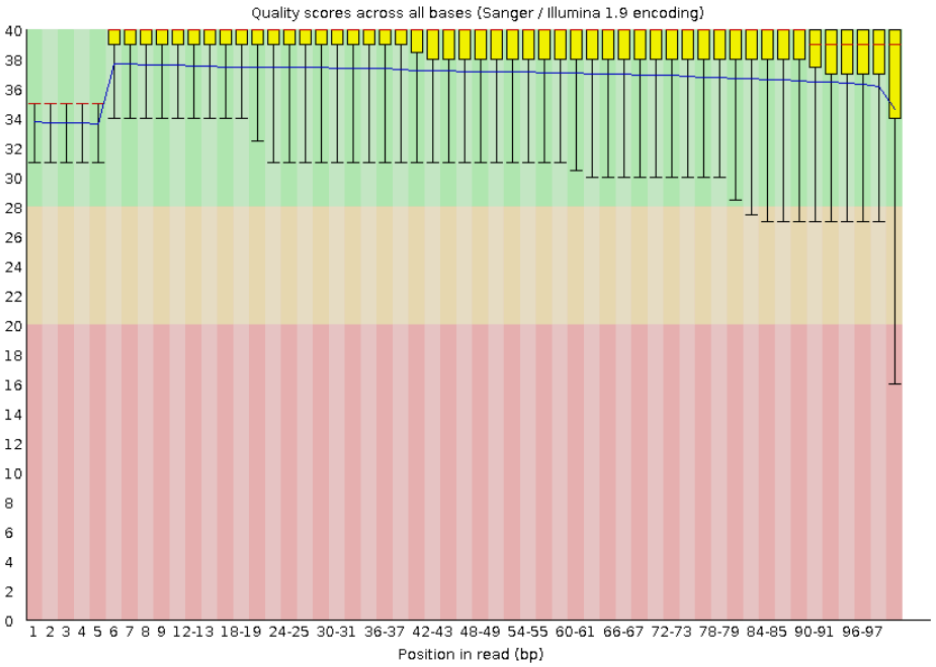

Рис 12. График качества одноконцевых чтений

Хмм, качество классное кроме начала (интересно что это, в экзмоном секе такого не видел) и конца (причём в самом конце прям плохо, красная зона, фу). А в остальном медианы больше 36 – кайф.

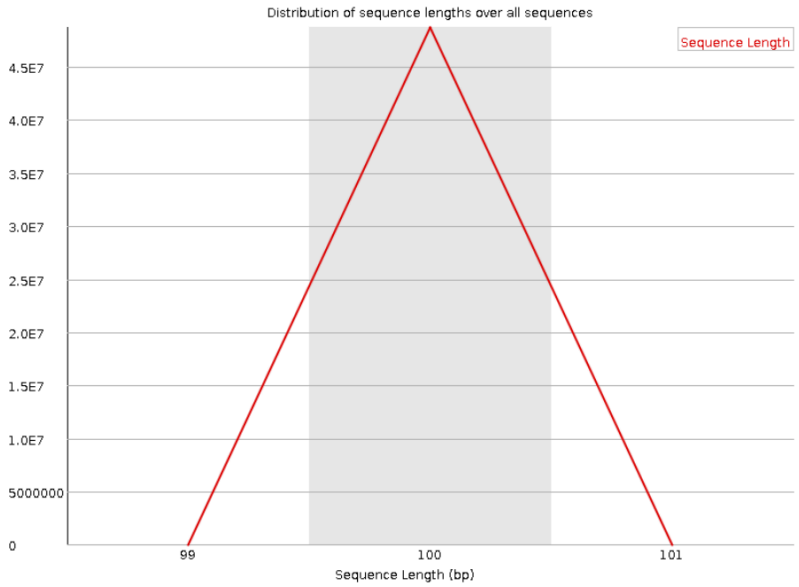

Рис 13. График распределения длины одноконцевых чтений

Длина всех чтений 100 п.о. – круть!
“И очутятся на бреге,
В чешуе, как жар горя,
Тридцать три богатыря,

Все красавцы удалые,
Великаны молодые,
Все равны, как на подбор,
С ними дядька Черномор.”
А. С. Пушкин

3. Картирование чтений на референс

Картирую чтения на уже проиндексированный референс (мою драгоценную 4-ю хромосому), а результат сохраняю в SAM-файл.

```
hisat2 -x chr4 -k 3 -U ENCFF641WPY.fastq.gz > RNA_map.sam
```

Команда: *hisat2* – программа для картирования чтений
Опции: *-x* – файл с индексацией референса
-k – количество самых качественных выравниваний
-U – файл с чтениями
Выдача: *RNA_map.sam* – файл с результатом

Логи команды:

```
48692928 reads; of these:
  48692928 (100.00%) were unpaired; of these:
    48667787 (99.95%) aligned 0 times
    23657 (0.05%) aligned exactly 1 time
    1484 (0.00%) aligned >1 times
0.05% overall alignment rate
```

На хромосому мою ненаглядную картировалось: *23657 + 1484 = 25 141* чтений или 0,0516% от общего кол-ва чтений. Сравнил с результатами однокурсников. МАЛО!!! Ну образец-то с мышиной РНК из сердца (одной конкретной ткани) был, а картирую я на человеческую хромосому, и к тому же все РНК клеток на лишь одну хромосому. Так что логично, что мало :-) Амир ответственно заявил с уверенностью на все 100%, что слишком мало и так не может быть. Не знаю, мне не с чем сравнить особо.

Далее перевел SAM в BAM:

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

Индексировал BAM-файл:

```
samtools index RNA_map.bam
```

Отобрал чтения, картированные на 4-ю хромосому:

samtools view -h -bS RNA_map.bam > RNA.bam

Все эти команды подробнее описано в предыдущих практикумах :-)

4. Поиск экспрессирующихся генов

В инструкции сказано, что мне очень нужен файл с генной разметкой. Я долго думал над этим, но в итоге не смог устоять перед столь заманчивым предложением! Изучил его.
Вот как он устроен:
1. Шапка: начинается с символа #, содержит общую информацию (версия разметки, версия генома, дата публикации генома, идентификатор разметки, дата последнего обновления).
2. Основные данные: 9 столбцов, включающих номер хромосомы, источник аннотации, тип особенности (ген, транскрипт, экзон), начало и конец гена, значение с плавающей запятой, цепь, на которой расположен ген, и дополнительная информация.

Дальше, хлебнув ещё немного чая, я для каждого гена из разметки посчитал число картированных на этот ген чтений.

htseq-count -f bam -s no -m union -t gene RNA_map.sam Homo_sapiens.GRCh38.110.chr.gtf > exp_RNA.txt

Команда: *htseq-count* – программа для подсчета этого самого, что нам надо (написано выше)
Опции: *-f bam* – формат входного файла
-s no – нет указания цепи (поэтому попадают и на прямую и на обратную)
-m union – **THE SOVIET UNION** (объединение перекрывающихся чтений)
-t gene – тип элементов с которыми считается перекрывания (в данном случае гены, потому что в задании о генах говорится). Хотя логично, что по умолчанию там экзоны смотрят, ибо у нас РНК-сек, а мРНК сплайсируется.
Выдача: *exp_RNA.txt* – файл с результатом

Справочная информация по названиям файлов, полезно для понимания:
seqcap_hg38.bed – координаты таргета (совокупность экзонов белок кодирующих генов)
seqcap_hg38_50.bed – координаты таргета (совокупность экзонов белок кодирующих генов), расширенные на 5 нукл с двух сторон

Конец файла exp_RNA.txt:
__no_feature 1159

__ambiguous 4186
__too_low_aQual 0
__not_aligned 48667787
__alignment_not_unique 1484

a) Кол-во чтений попало в границы генов: *4186 + 1484 = 5670*
b) Кол-во чтений попало мимо границ генов: *1159* (__no_feature)

5. Аннотация высоко экспрессируемых генов

Я произвел сортировку второго столбца файла, содержащего экспрессионный профиль, который был получен на предыдущем этапе работы. В результате сортировки данные в этом столбце были упорядочены в порядке убывания, что позволяет более наглядно увидеть, какие значения являются наибольшими, а какие — наименьшими. Это упрощает анализ и интерпретацию полученных результатов, так как теперь информация представлена в более структурированном виде.

```
cat exp_RNA.txt | head -n -5 | sort -k2nr > exp.txt
head -n 10 exp.txt > exp_sort.txt
```

Топ 10 самых высоко экспрессируемых генов:
ENSG00000109332 4087
ENSG00000151729 2197
ENSG00000132466 904
ENSG00000138670 693
ENSG00000138668 490
ENSG00000151247 484
ENSG00000109819 438
ENSG00000109606 395
ENSG00000138767 348
ENSG00000207322 322

Настал час выбрать ген. Я выбрал *ENSG00000109332 4087*, ибо он наиболее экспрессируемый и белок-кодирующий.

Далее я визуализировал этот ген и его консервативность в геномном браузере. Привожу изображение, на котором изображены экзон-интронные структуры известных для данного гена транскриптов. Всего транскриптов отобразилось 15 штук. Видно, что экзоны довольно маленькие в сравнении с интронами, что в целом характерно для организации генома высших эукариот.

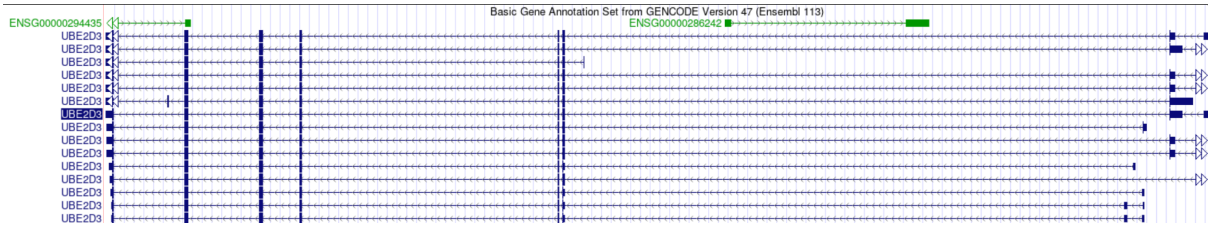

Рис 14. Экзон-интронные структуры известных транскриптов для гена ENSG00000109332

Если обратить внимание на треки консервативности, то можно заметить, что наиболее консервативные участки – это экзоны, что логично, ведь там кодирующая белковый продукт последовательность, которая наиболее чувствительна к мутациям (белок может потерять функцию). Но ещё надо заметить, что есть несколько относительно консервативных участков и в экзонах. Думаю, они важны для сплайсинга или для регуляции транскрипции этих РНК.

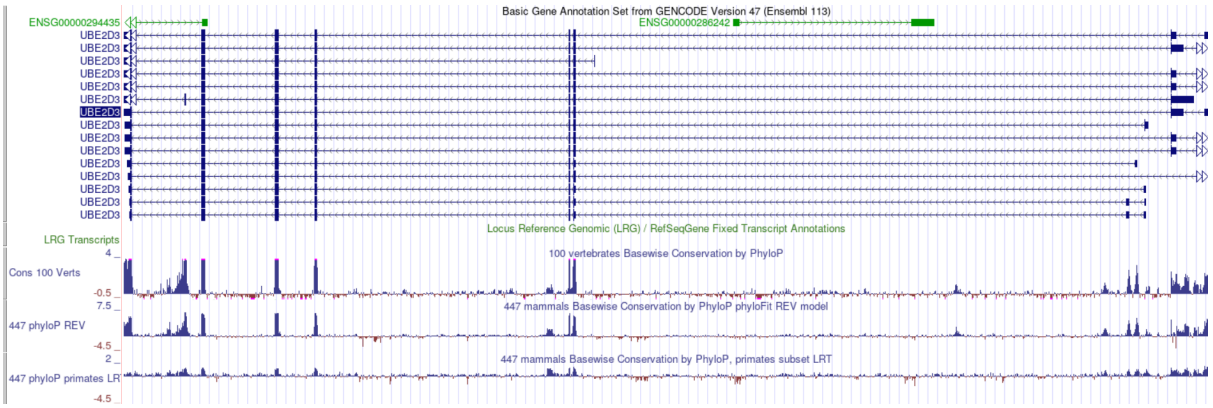

Рис 15. Трек консервативности

Рассмотренный ген кодирует белок, который является частью семейства убиквитин-конъюгирующих ферментов (Ubiquitin Conjugating Enzyme E2 D3). Этот белок играет важную роль в процессе убиквитинирования, который регулирует деградацию белков. В частности этот фермент функционирует в убиквитинировании белка-супрессора опухолей p53, который является важным регулятором клеточного цикла.

СКРИПТ

Путь к файлу: ~public_html/term3/script.sh

```
#!/bin/bash

# На ввод подается имя хромосомы в формате цифры: N example "9"
#На ввод подается ID чтения в формате: SRRXXXXXXXX example
"SRR10720410"

#chr=$1
#reads=$2
chrom=/mnt/scratch/NGS/DATA/dna_reads
raw_reads=/mnt/scratch/NGS/DATA/dna_reads
ref=/mnt/scratch/NGS/DATA/genes

hisat2-build $chrom/Homo_sapiens.GRCh38.dna.chromosome.$1.fa chr$1
#индексация референса, chr$1 - префикс имен создаваемых файлов

samtools faidx $chrom/Homo_sapiens.GRCh38.dna.chromosome.$1.fa
#индексация для программы samtools

fastqc $2_1.fastq.gz
fastqc $2_2.fastq.gz
#проверка качества чтений: прямых и обратных

TrimmomaticPE -phred33 $raw_reads/$2_1.fastq.gz
$raw_reads/$2_2.fastq.gz tr1_pair.fastq.gz tr1_unpair.fastq.gz
tr2_pair.fastq.gz tr2_unpair.fastq.gz TRAILING:20 MINLEN:50
#TrimmomaticPE - Trimmomatic для парноконцевых чтений (PE),
фильтрация чтений
#-phred33 - quality score
#TRAILING:20 - удаление нуклеотидов с конца с качеством ниже 20
#MINLEN:50 - удаление чтений с длиной меньше 50

hisat2 -x chr$1 -1 tr1_pair.fastq.gz -2 tr2_pair.fastq.gz -p 4
--no-spliced-alignment > chr$1_map.sam 2> logs.txt
#картирование чтений на референс
# -x - префикс имен файлов, которые были получены при индексации
#-1 tr1_pair.fastq.gz - файл с прямыми оттриммамаченными
парными чтениями
#-2 tr2_pair.fastq.gz - файл с обратными оттриммамаченными
```

```
парными чтениями
#-p 4 - использование 4 ядер процессора (вроде, это доступный максимум (?))
#--no-spliced-alignment - запрет на сплайсирование, то есть на картирование с разрывами
#> chr9_map.sam - запись результатов в этот файл
#2> logs.txt - сохранение логов в txt файле

samtools sort -o chr$1_map.bam chr$1_map.sam
#Конвертации sam в bam

samtools index chr$1_map.bam
#Индексация полученного bam файла

samtools flagstat chr$1_map.bam > ch$1_flagstat.txt
#Анализ bam файла

samtools view -h -bS chr$1_map.bam $1 > $1chr.bam
#Чтения, картированные только на данную хромосому
#-f 2 - вывод только тех чтений,  которые точно выровнены с референсом (прошли по критерию, соответствуют PROPER_PAIR)
#-bS = = -b (вывод в формате bam) + -S (автоматическое определение формата подаваемого файла)

samtools flagstat $1chr.bam > $1ch_flagstat.txt
#Заглядывание в получившийся файл

samtools view -f 2 -bS $1chr.bam > true_pairs.bam
#Получение только правильно картированных пар чтений
#-f 2 - вывод только тех чтений,  которые точно выровнены с референсом (прошли по критерию, соответствуют PROPER_PAIR)
#-bS = = -b (вывод в формате bam) + -S (автоматическое определение формата подаваемого файла)

samtools flagstat true_pairs.bam > true_flagstat.txt
#Заглядывание в получившийся файл

samtools index true_pairs.bam
#Индексация полученного bam файла

bedtools intersect -a true_pairs.bam -b $ref/seqcap_hg38.bed > bed_true.bam
```

```
#Получение чтений, картированных только в границы экзома

samtools flagstat bed_true.bam > bed_flagstat.txt
#Заглядывание в получившийся файл

bcftools mpileup -f Homo_sapiens.GRCh38.dna.chromosome.9.fa
true_pairs.bam | bcftools call -mv -o true_pairs.vcf

#bcftools mpileup - создает vcf файл с вероятностями
#bcftools call - выбирает только указанные строки из выдачи
предыдущей программы
#-f - указание на референс
#-mv - поиск мультиаллельных и редких вариантов, выводит их
#-o - output

bcftools stats true_pairs.vcf > stats.txt
#Анализ vcf файла

bcftools filter -i 'QUAL>30 && DP>50' true_pairs.vcf -o var.vcf
#Фильтрация вариантов по качеству и покрытию

bcftools stats var.vcf > var.txt
#Анализ vcf файла
```