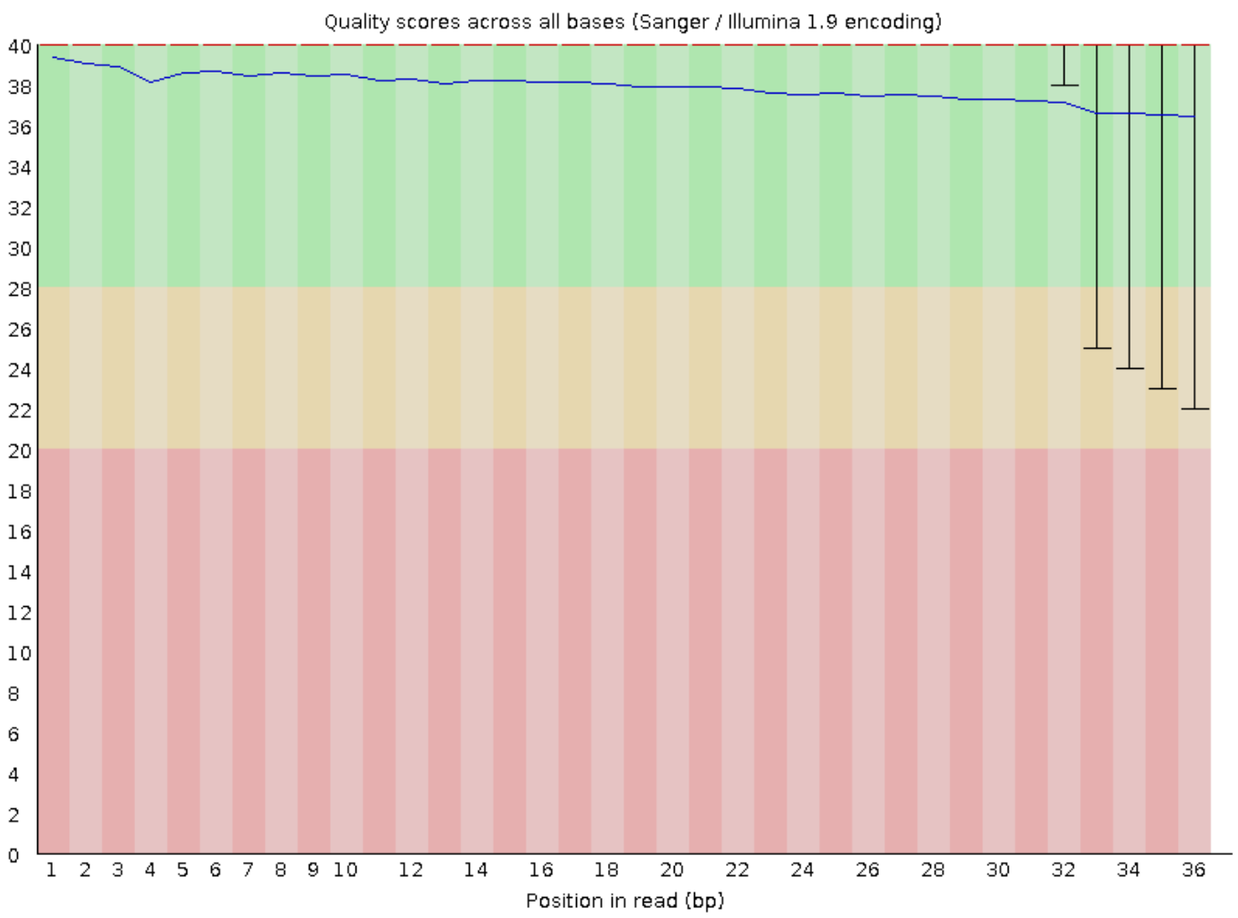
Term 3, Pr 15, Genome assembly

Осмотримся (`fastqc`)

Сперва посмотрим на исходное качество чтений, начнём с *per base quality score*:



Видим, что данные уже неплохие, и, по всей видимости, большая триммирования будет состоять в отрезании нескольких нуклеотидов с конца некоторых последовательностей.
Кроме того, у нас ест некоторое количество перепредставленных последовательностей:

Sequence	Count	Percentage	Possible Source
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	27619	0.6247812790473302	No Hit
GATCGGAAGAGCTCGTATGCCGTCTTCTGCTTGAAA	14021	0.31717507199835676	Illumina Single End Adapter 1 (100% over 33bp)
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	11336	0.2564365320714195	No Hit
ANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	11116	0.25145981744053447	No Hit
TNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	10528	0.23815841651798733	No Hit
GATCGGAAGAGCTCGTATGCCGTCTTCTGCTTAGAT	10033	0.22696080859849607	Illumina Single End Adapter 1 (96% over 32bp)
GATCGGAAGAGCTCGTATGCCGTCTTCTGCTTGGAT	9797	0.2216221510853649	Illumina Single End Adapter 1 (100% over 33bp)
GNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	5331	0.12059484407839953	No Hit

Запомним, что до триммирования у нас было 4420587 чтений и весили они 94,5 Мб.

`trimmomatic`

Поскольку мы хотим удалять адаптеры, нам нужен файл с ними, сделаем его из всего, что есть в папке `adapters` и запустим оттуда следующий скрипт:

```
cat * > ../a.k.rybakov/assembly/adapters.fasta
```

Теперь выполним триммирование:

```
TrimmomaticSE SRR4240378.fastq.gz trimmed.fq.gz
ILLUMINACLIP:adapters.fasta:2:7:7 TRAILING:20
MINLEN:32 -threads 20 -trimlog trimming.log
```

Триммирование сохранило 4154738 (93,99%) от всех чтений. Но поскольку нам ещё интересно, сколько именно чтений удаляется из-за наличия в них адаптерных последовательностей, сделаем "боковой"

прогон `trimmomatic`'а - боковой в том смысле, что файлы из него никуда дальше не пойдут:

```
TrimmomaticSE SRR4240378.fastq.gz
partially_trimmed.fq.gz
ILLUMINACLIP:adapters.fasta:2:7:7 -threads 20 -
trimlog parial_trimming.log
```

Этот запуск даёт понять, что именно из-за связи с адаптерами мы бракуем 81843 (1,85%) ридов.

`velveth` и `velvetg`

Сначала построим библиотеку kmer'ов с помощью `velveth`:

```
velveth kmers 31 -short -fastq trimmed.fq.gz >
velveth.log
```

Что насобиралось?
Ответим на этот вопрос, поработав с файлом `stats.txt`, который теперь появился в нашей папке `kmers`:

```
sort -k2rn stats.txt > top_len.txt #по длине
контига в порядке невозрастания
sort -k6rn stats.txt | less #по средему покрытию
```

Объект	Значение
N50	7028
Длина первого по длине контига	36746
Длина второго по длине контига	19371
Длина третьего по длине контига	16745

Типичным покрытием для контига в этом эксперименте мы можем считать 20, однако можно отыскать такие,

покрытие которых находится в пределах от 200 до 900, есть так же контиг, покрытый в среднем 148170 раз (тут нас должно было напрячь, что это число не дробное). И, внезапно, мы понимаем, в чём же дело: большинство этих контигов имеют длину 1, в редких случаях не больше 7. Они несодержательны, поэтому попробуем найти контиги с интересным покрытием, длина которых хотя бы больше размера хэша (то есть 31):

```
sort -k2rn stats.txt |head -n 200 | sort -k6rn > top_cov.txt
```

Находим два аномальных контига:

Длина	Покрытие
934	102,75
2106	100,56

Трудно сказать про эти контиги что-то особенное: длину и покрытие мы указали, остаётся только последовательность. Довольно опрометчиво делать какие-то выводы на глаз, но можно, например, сказать, что обе последовательности заканчиваются достаточно большим количеством A, а у более короткой много A и в начале. Кроме того, примерно одинаково большое покрытие могло бы, очевидным образом, объясняться схожестью последовательностей этих контигов, но локальное выравнивание не обнаруживает значимых находок.

BLAST: выравнивание контигов с геномом

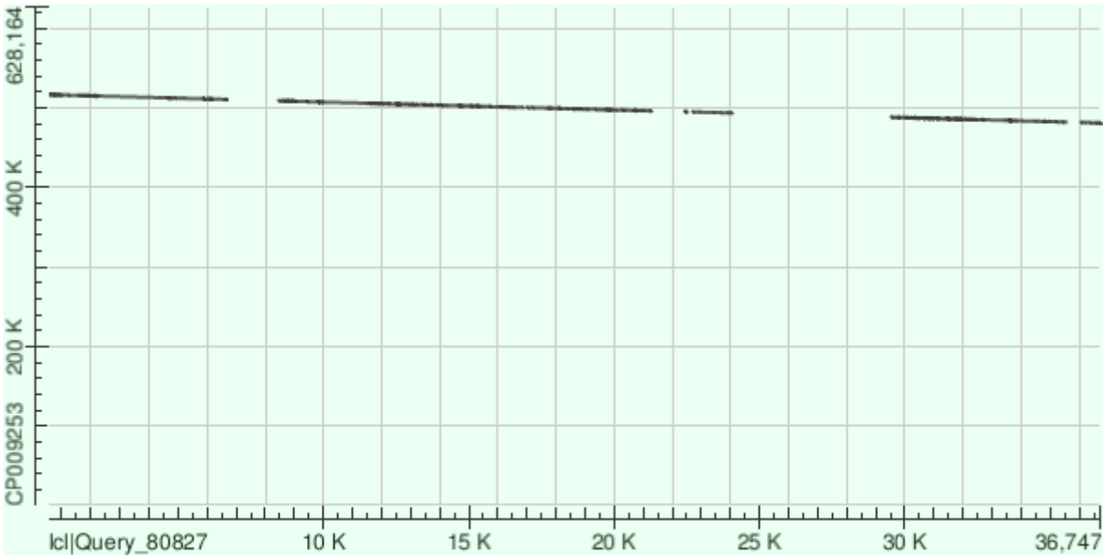
Будем ровнять с геномом три самых длинных контига. Для каждого контига будем рисовать DotPlot, графически

отображать координаты на хромосоме и выводить некоторую информацию о полученных выравниваниях (здесь координаты возьмём геномные).

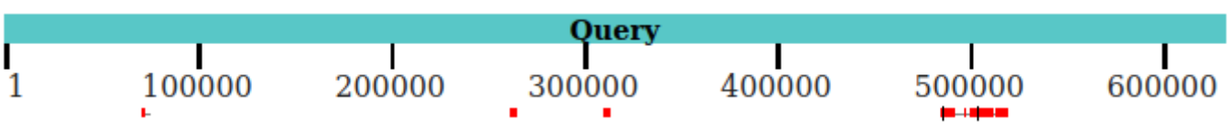
По дотплотам из-за различий в масштабах осей будет сложно сделать выводы, например, о двойном картировании (близкие параллельные линии), но вот разрывы будет видно прекрасно.

Контиг длины 36746

Тут мне пришлось получить транспонированный дотплот, т. к. обычный не строился. Для двух других контигов линия будет почти вертикальной.



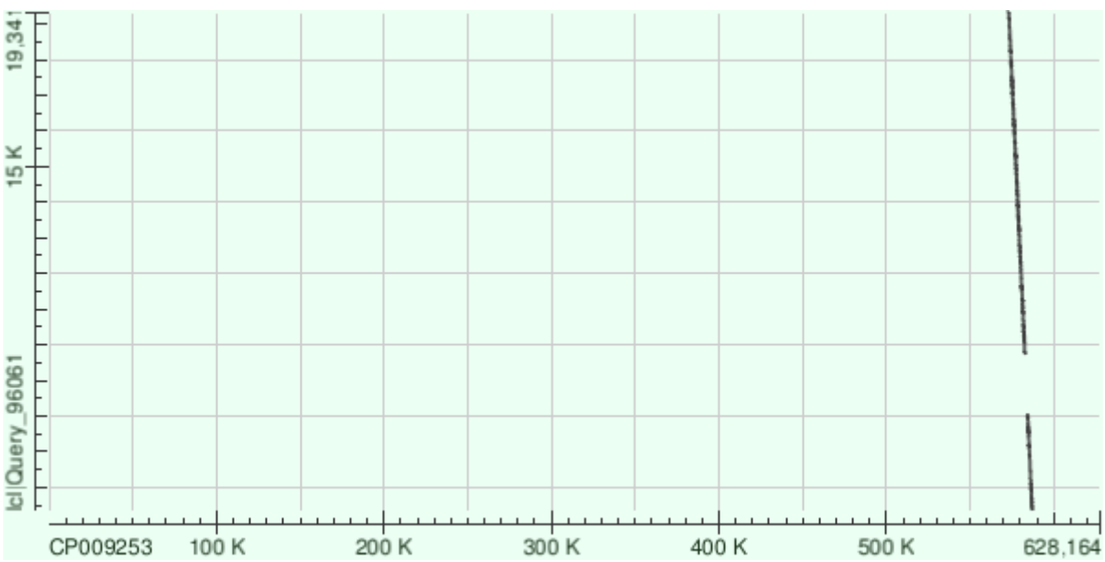
Видим, что по версии авторов сборки генома *Buchnera aphidicola*, с которой мы сравниваемся, наш контиг, на самом деле, только скэффолд - нашлись кусочки генома, разделившие нашу последовательность.



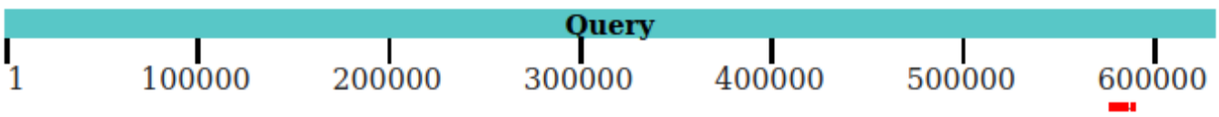
Выравнивание	Начало	Конец	Длина	Доля гэпов
1	500370	508984	8614	0,04
2	510441	516675	6234	0,03
3	481997	488261	6264	0,05
4	496111	500436	4325	0,03

Выравнивание	Начало	Конец	Длина	Доля гэпов
5	493487	494872	1385	0,01
6	480874	481561	687	0,02

Контиг длины 19371

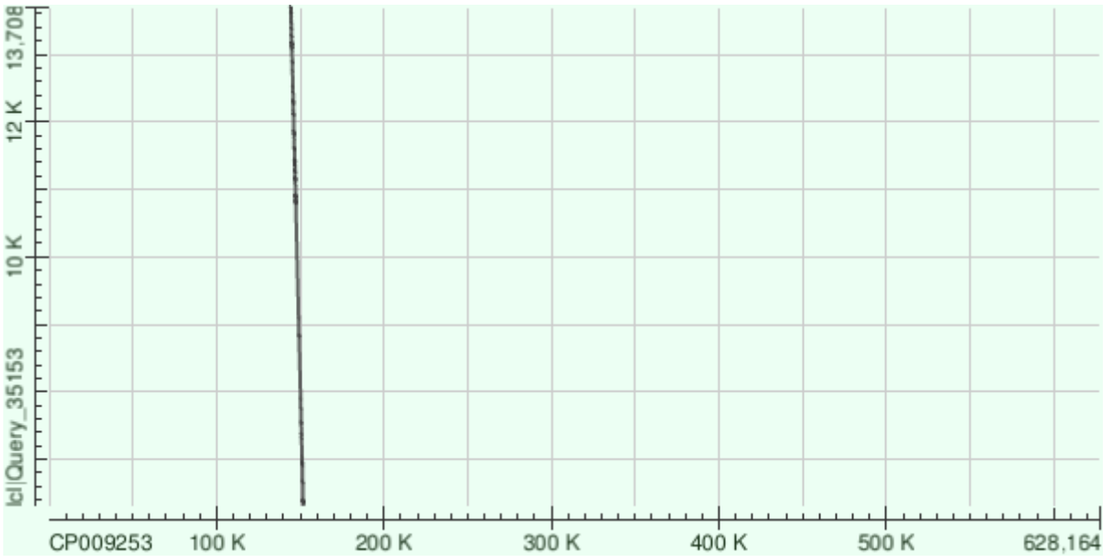


Можем сделать такой же вывод, с предшествующим контигом - по версии сборки `CP009253` наша последовательность контигом не является, но если теперь уточнить координаты разрыва, то получится в точности скэффолд)

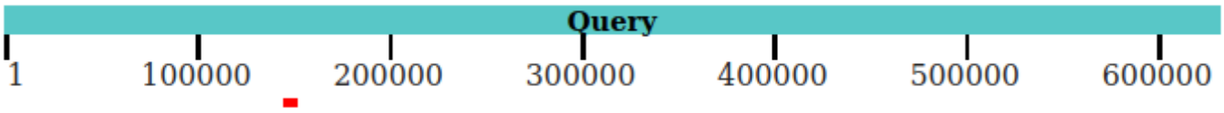


Выравнивание	Начало	Конец	Длина	Доля гэпов
1	573092	592493	19401	0,04
2	584329	587107	2778	0,03

Контиг длины 16745



А вот здесь мы видим, что наш контиг, скорее всего, непрерывен и по версии авторов сборки генома!



Выравнивание	Начало	Конец	Длина	Доля гэпов
1	144368	151906	7538	0,03

С другой стороны, мы видим, что небольшая доля гэпов есть, совпадение не полное. Связано ли это с техническими различиями, или геном собирался на данных другого секвенирования и различия содержательны? Второй вариант вероятнее.

Мысль по поводу гэпов

Я думаю, что печатать доли гэпов для каждого выравнивания здесь имело мало смысла, поскольку в нашей ситуации это число скорее определялось параметрами `megablast`'а, чем особенностями выравнивания.