Геномные интервалы. Формат BED

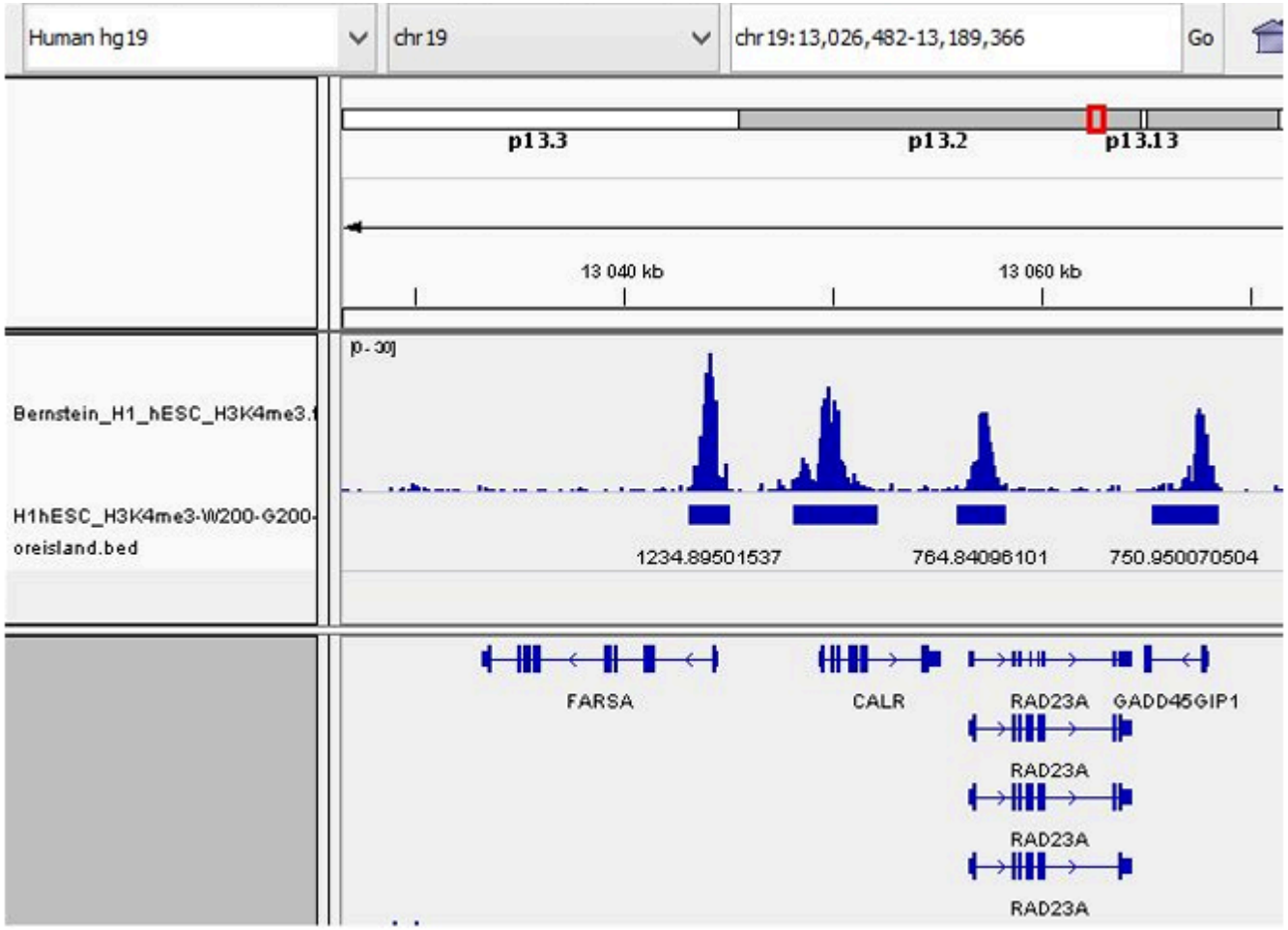
зачем используются геномные интервалы какие варианты бывают у используемого для представления интервалов формата BED, какие операции можно проделывать с геномными интервалами при помощи bedtools

Page • Tag • 1 backlink

Геномные интервалы

- универсально описывают участки генома в координатных последовательностях

- экзоны, интроны, промоторы, энхансеры, повторы и прочее можно выделить в виде интервала
- видим в каждом гене: начало, направление транскрипции, экзоны (толстые), интроны между, конец
- верхний трек с гистограммами: покрытие, прямоугольники — интервал покрытия



Формат BED

1. 3 необходимые колонки (\t)
2. до 9 опциональных колонок

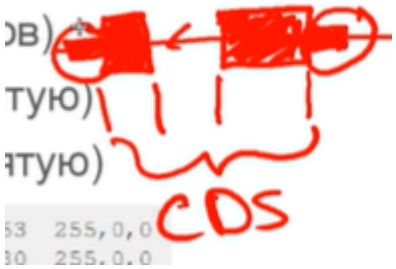
3. 0-based: длина интервала

   = *chrEnd - chrBegin*

Формат BED подразумевает нумерацию последовательности с 0 и не включает нуклеотид справа. Таким образом, интервал [0,5) содержит 5 нуклеотидов

Расширенный BED

- name - имя
- score - число, 0..1000
- trand - нить ДНК (+ или -)
- thickStart,thickEnd - начало/конец широкого интервала (маленькие прямоугольники это UTR)



- itemRgb - цвет в RGB (e.g. 0,0,255)
- blockCount - количество суб-элементов (экзонов)
- blockSizes - размер суб-элементов (через запятую)

- blockStarts - начало суб-элементов (через запятую)

```
chr7  127471196  127472363  Pos1  0  +  127471196  127472363  255,0,0
chr7  127472363  127473530  Pos2  0  +  127472363  127473530  255,0,0
chr7  127473530  127474697  Pos3  0  +  127473530  127474697  255,0,0
chr7  127474697  127475864  Pos4  0  +  127474697  127475864  255,0,0
chr7  127475864  127477031  Neg1  0  -  127475864  127477031  0,0,255
chr7  127477031  127478198  Neg2  0  -  127477031  127478198  0,0,255
chr7  127478198  127479365  Neg3  0  -  127478198  127479365  0,0,255
chr7  127479365  127480532  Pos5  0  +  127479365  127480532  255,0,0
chr7  127480532  127481699  Neg4  0  -  127480532  127481699  0,0,255
```

Строка track

- позволяет разделить BED на несколько трэков
- позволяет задавать их номера, цвет, и тип

```
browser position chr7:127471196-127495720
track name="TilingArray" description="TilingArray demonstration" visibility=2 useScore=1 height=30
7  127471196  127472363  Pos1  700  +  127471196  127472363  255,0,0
7  127472363  127473530  Pos2  800  +  127472363  127473530  255,0,0
7  127473530  127474697  Pos3  900  +  127473530  127474697  255,0,0
```

```
browser position chr7:127471196-127495720
track name="Histogram" description="Histogram demonstration" visibility=2 useScore=3 height=30
7  127471196  127472363  Pos1  700  +  127471196  127472363  255,0,0
7  127472363  127473530  Pos2  800  +  127472363  127473530  255,0,0
7  127473530  127474697  Pos3  900  +  127473530  127474697  255,0,0
```

Bedtools

https://quinlanlab.org/tutorials/bedtools.html

- программа для работы с интервальными данными
- сортировка, пересечение, статистика и тд

Команды

bedtools sort - сортировка

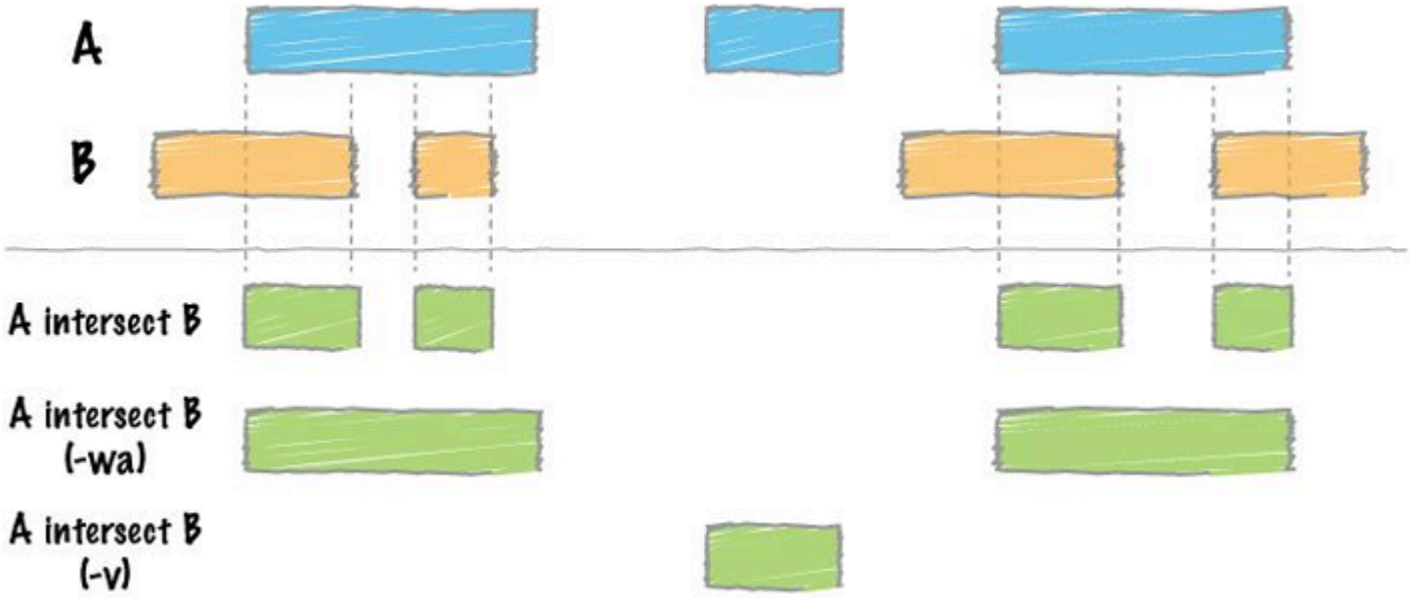
- хромосомы по алфавиту, координаты по возрастанию
- числа: chr1,chr10,chr11,...,chr2,chr21,chr22,chrM,chrX,chrY

- сортировка ускоряет многие операции

```bash
bedtools sort -k1,1 -k2,2n file.bed > file.sorted.bed
```

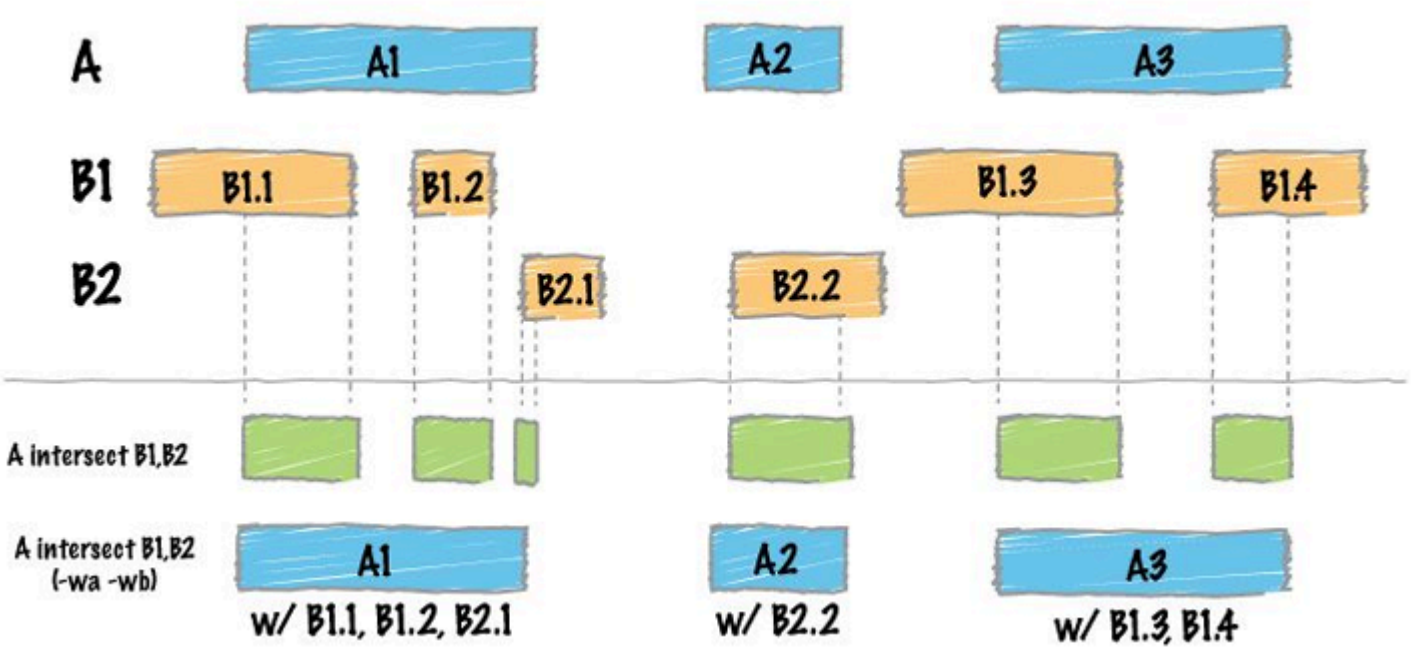
Пересечение интервалов

можно оставить оригинал (A,B), подсчитать нуклеотиды в пересечении, найти только непересекающиеся, и т.д.

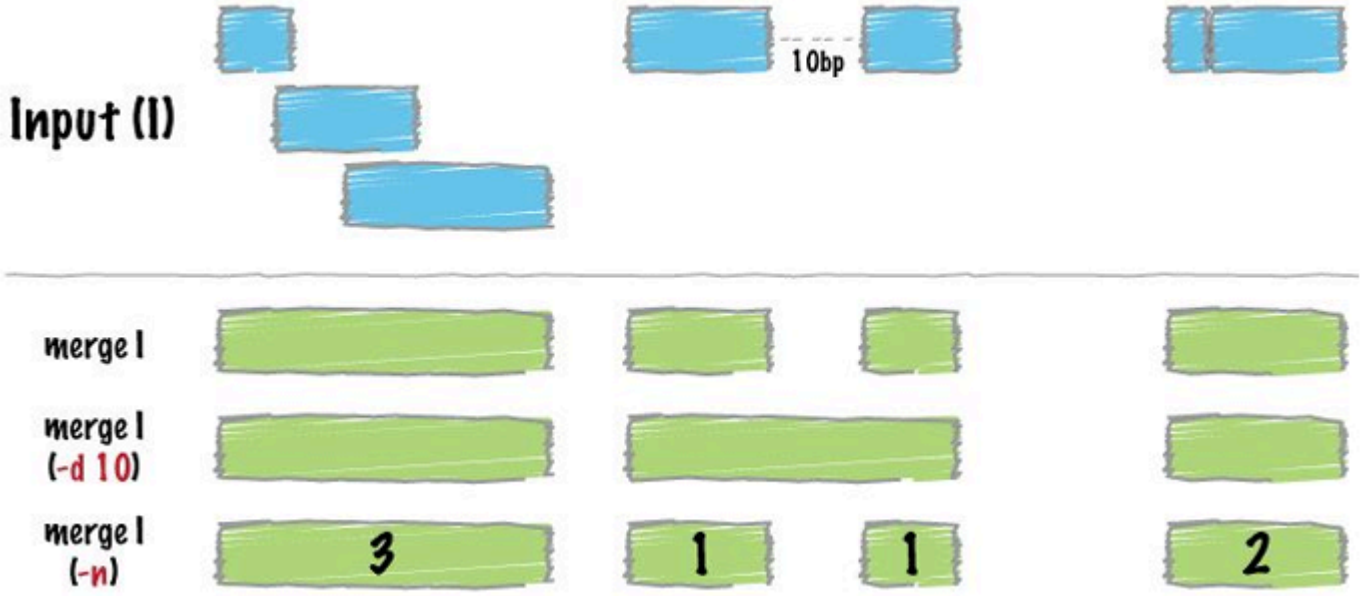


-wa все, включение из A, -v что не включилось в интервал

bedtools выше 2.21.0



Объединение



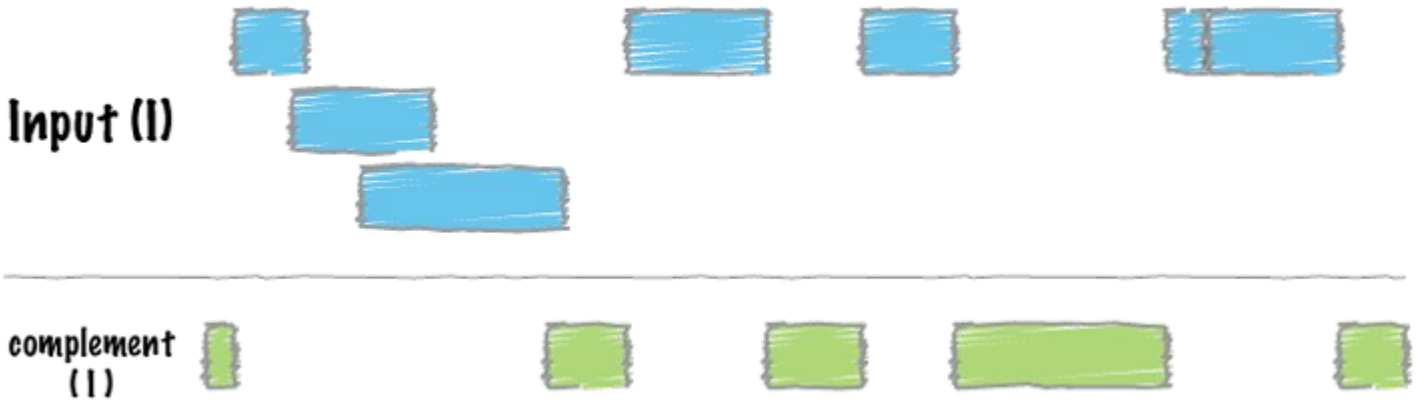
merge I — все объединить по границе или наложении в интервал

merge I (-d 10) — с пересечением до 10 п.н.

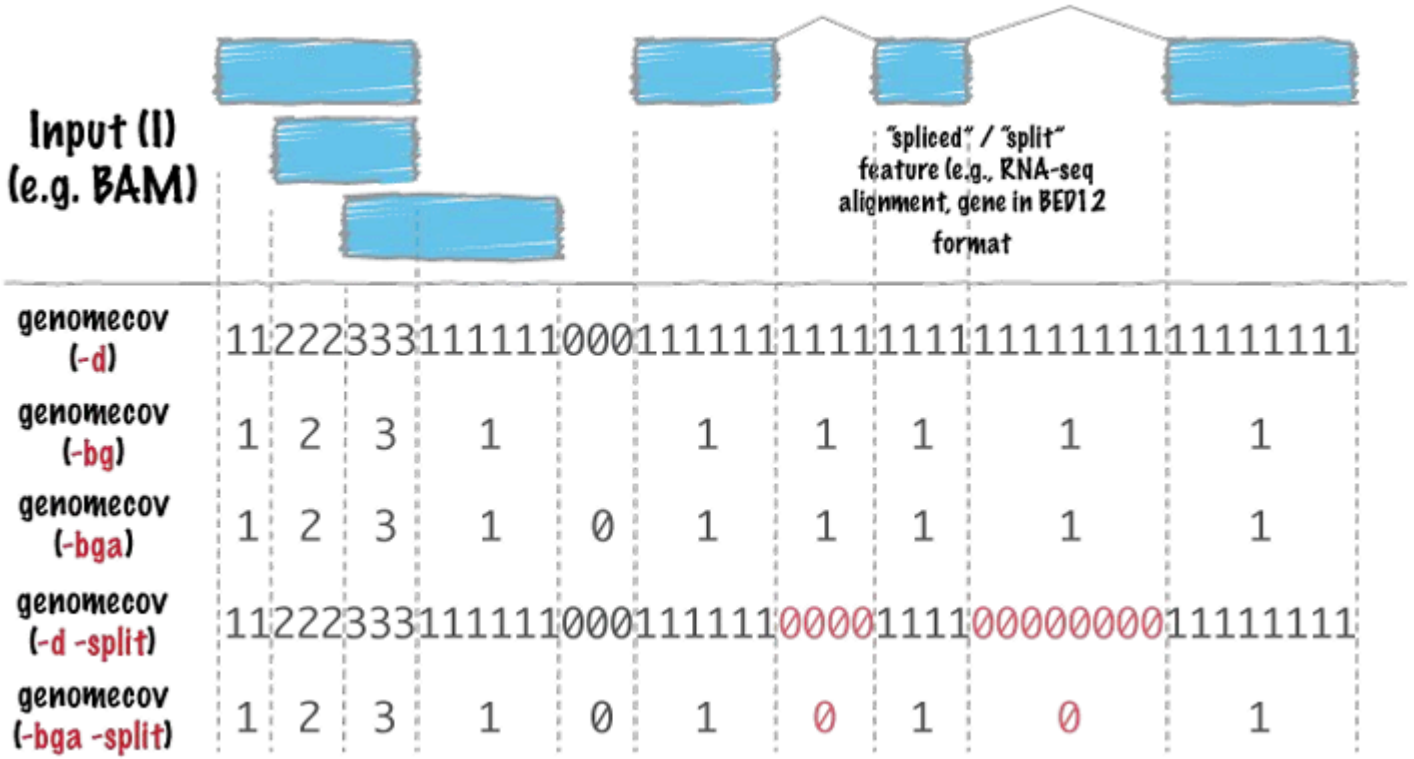
merge I (-n) — по количеству лежащих рядом

Вычет (комплемент)

- находит регионы, не покрытые входящими интервалам



Расчет покрытия



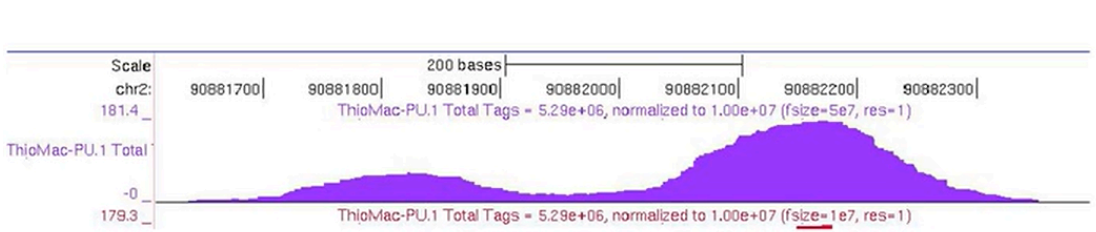
-d подсчет покрытия на каждый нукдеотид

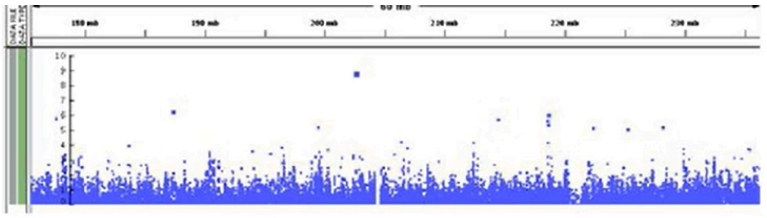
-bg -bga интервальный расчет в единый (bga учитывает нули)

-d -split прочтения на геном из транскриптома (экзон-интронную структуру позволяет учитывать)

BEDgraph

- визуализация - прямоугольники или точки





```
track type=bedGraph name="BedGraph Format" description="BedGraph format" priority=20
chr19 59302000 59302300 -1.0
chr19 59302300 59302600 -0.75
chr19 59302600 59302900 -0.50
chr19 59302900 59303200 -0.25
chr19 59303200 59303500 0.0
chr19 59303500 59303800 0.25
chr19 59303800 59304100 0.50
chr19 59304100 59304400 0.75
```

"priority bedgraph" — это заранее заданный порядок треков, если их больше одного. т.е. можно прописать от 1 до 20, и браузер выстроит их в нужном порядке

BEDOPS

bedops.readthedocs.io

BEDOPS: the fast, highly scalable and easily-parallelizable genome analysis toolkit — BEDOPS v2.4.41

BEDOPS: the fast, highly scalable and easily-parallelizable genome analysis toolkit¶ BEDOPS is an open-source command-line toolkit that performs highly efficient and scalable Boolean and other set ope...

1. быстрые и хорошо параллелизуемые программы
2. упор на очень большие файлы

Задача

💡 Скачайте аннотацию генома *D. melanogaster* от консорциума RefSeq, и распакуйте ее. Полученный файл должен содержать 12 колонок, разделенных знаками табуляции. Идентификаторы транскриптов в 4-й колонке должны начинаться с символов *NM_* (в случае протеин-кодирующих транскриптов) или *NR_* (в случае некодирующих транскриптов). Используя полученный файл, а также консольные команды **grep**, **cut**, **sort**, **wc**, и **uniq** (и их комбинирование при помощи **pipe**), определите, сколько уникальных протеин-кодирующих и некодирующих транскриптов присутствует в данной аннотации. Введите числа в указанном порядке через запятую, без пробела.

Plain Text

```
cut -f 2 file.txt  | sort | uniq | wc -l
```

- grep 'NM' — выводит все строки файла, содержащие данную подстроку
- wc - l считает число строк в файле
- uniq удаляет из файла одинаковые строки, но ТОЛЬКО ЕСЛИ ОНИ ИДУТ ПОДРЯД. Поэтому нам нужно сортировать файл перед применением этой команды при помощи sort
- поскольку в задании просят найти число уникальных транскриптов, и приведён пример со второй строкой, возникает желание считать одинаковыми те транскрипты, которые начинаются с одной позиции. На самом деле одинаковыми следует считать транскрипты, имеющие одно и то же имя
- вместо cut можно использовать awk '{print $<column_number>}'

💡 Используя полученный ранее 12-колоночный BED-файл с аннотацией генома *D. melanogaster* от RefSeq, превратите его в 6-колоночный BED, в котором каждый интервал будет соответствовать индивидуальному экзону из аннотации. Для конвертации используйте команду **bedtools bed12tobed6**. Далее, при помощи

команды **grep** сосчитайте количество экзонов, приходящихся на (+)-нить ДНК, на (-)-нить ДНК, а также их общее количество. Введите три числа в указанном порядке, через запятую и без пробелов

Plain Text

```
bedtools bed12tobed -i dm6_refseq.bed > dm6.bed
grep  -e '+' dm6.bed | cut -f 6 | sort | uniq -D | wc -l
grep  -e '-' -e '+' dm6.bed | cut -f 6 | sort | uniq -D | wc -l
```