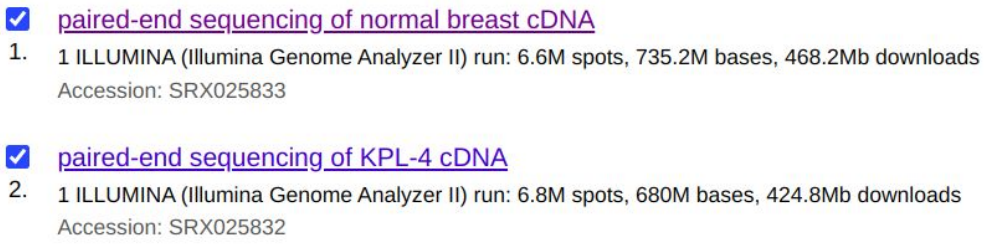
ДЗ номер 3 по сборке транскриптома.
Чангалиди Антон

Task 2020
Dataset: https://www.ncbi.nlm.nih.gov/sra/?term=srp003186
Возьму два семпла (один - рак, другой - обычный):



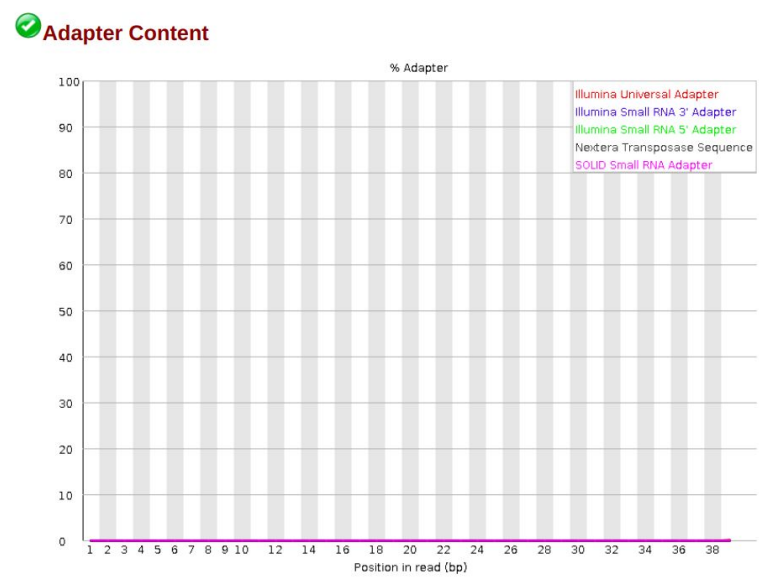
Acession numbers:
SRR064287 (cancer)
SRR064437 (normal)

```
mkdir data
fastq-dump -split-3 -O data/ --gzip SRR064287
fastq-dump -split-3 -O data/ --gzip SRR064437
```

0. хочу проверить качество, и запустить обычный триммоматик.

```
fastqc -o data/ data/*.fastq.gz
```

1. Run adapter-trimming software of your choice to get clean reads
Fastqc показывает, что адаптеров нет (те скачки контента в начале - возможно просто сайты рестрикции)**:**



Значит, ничего резать не надо. Окей, переходим к следующему пункту.

2. **Run rnaSPAdes** (http://cab.spbu.ru/software/rnaspades/) to assemble the combined transcriptome of normal tissues and a cancer cell line of your choice **Буду использовать rnaSPADES, так как есть небольшой опыт.** После установки запустим:

```
rnaspades -1 data/SRR064437_1.fastq -2 data/SRR064437_2.fastq
-m 10 -t 8 -o out_normal/
rnaspades -1 data/SRR064287_1.fastq -2 data/SRR064287_2.fastq
-m 10 -t 8 -o out_cancer/
```

3. run cd-hit-est (http://weizhongli-lab.org/cd-hit/) to cluster similar transcripts
Запускается так: ~/tools/cdhit/cd-hit-est

```
~/tools/cdhit/cd-hit-est -i out_normal/transcripts.fasta -o
out_normal/hit_est_output
~/tools/cdhit/cd-hit-est -i out_cancer/transcripts.fasta -o
out_cancer/hit_est_output
```

Получил что-то, пойдем дальше.

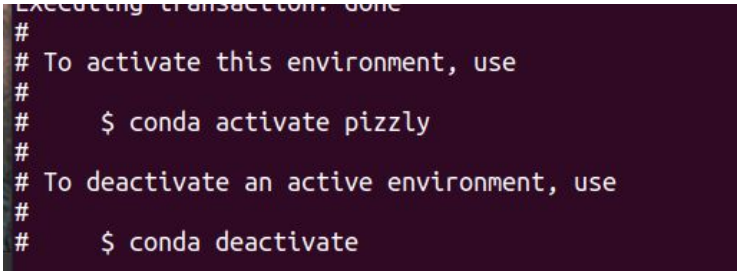
4. Annotate with blast+ (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/) using Uniprot SwissProt (can be downloaded here https://www.uniprot.org/downloads) as a reference

Тут надо было разбираться, я решил не доделывать эти шаги, и приступить к 5-7.

5. Find fusion transcripts with pizzly (https://github.com/pmelsted/pizzly)

Скачаю референсы:

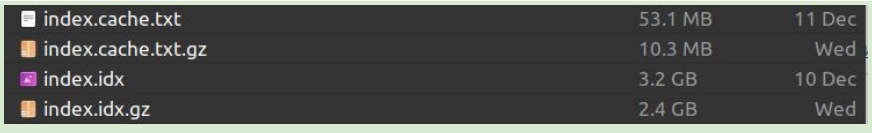
```
wget
ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/releas
e_36/gencode.v36.annotation.gtf.gz
wget
ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/releas
e_36/gencode.v36.transcripts.fa.gz

# не забудь, что надо палочки в аннотации заменить
zcat gencode.v36.transcripts.fa.gz  | tr '|' ' ' | gzip -1 >
gencode.v36.transcripts.fixed.fa.gz
# и индексы для каллисто строить по fixed референсу!
```



conda activate pizzly

 # First we create the kallisto index

```
kallisto index -i index.idx -k 31 transcripts.fa.gz

(этот файл я скачал у Ларисы - их должно быть 2):
```

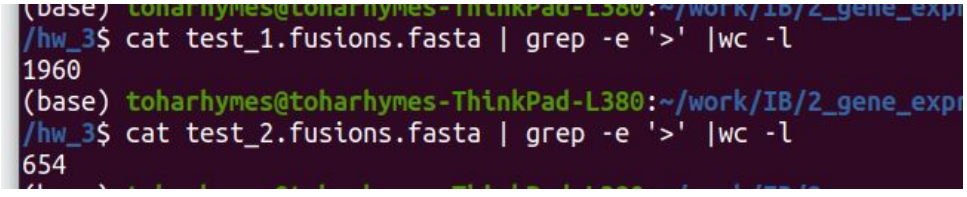


Next we quantify using kallisto with fusion detection enabled

```
kallisto quant -i ./references/index.idx --fusion -o output_1
data/SRR064287_1.fastq.gz data/SRR064287_2.fastq.gz

pizzly -k 31 --gtf ./references/gencode.v36.annotation.gtf
--cache test_1.index.cache.txt --align-score 2 --insert-size
400 --fasta references/gencode.v36.transcripts.fa.gz --output
test_1 output_1/fusion.txt

kallisto quant -i ./references/index.idx --fusion -o output_2
data/SRR064437_1.fastq.gz data/SRR064437_2.fastq.gz

pizzly -k 31 --gtf ./references/gencode.v36.annotation.gtf
--cache test_2.index.cache.txt --align-score 2 --insert-size
400 --fasta references/gencode.v36.transcripts.fa.gz --output
test_2 output_2/fusion.txt

conda deactivate
```

Итак, у нас были датасеты:

SRR064287 (cancer)
SRR064437 (normal)
Поэтому файлы _1 - раковые, _2 - обычные.



Фьюжнов в 3 раза меньше в обычных клетках. makes sense.

Теперь воспользуемся json-файлом и командами:

```
pizzly_flatten_json.py test_1.json cancer_fusion.txt
pizzly_flatten_json.py test_2.json normal_fusion.txt
```

И отфильтруем полученные транскрипты (чтобы не ноль), получим:

Обычный транскрипт:

	geneA.name	geneB.name	paircount	splitcount
24	HLA-C	HLA-B	87	9
47	CCDC32	CBX3	2	1
27	NPEPPS	TBC1D3D	1	3
28	NPEPPS	TBC1D3I	1	3
29	NPEPPS	TBC1D3L	1	3
30	NPEPPS	TBC1D3K	1	3
31	NPEPPS	TBC1D3	1	3
32	NPEPPS	TBC1D3G	1	3
33	NPEPPS	TBC1D3E	1	3
34	NPEPPS	TBC1D3C	1	3
35	NPEPPS	TBC1D3H	1	3

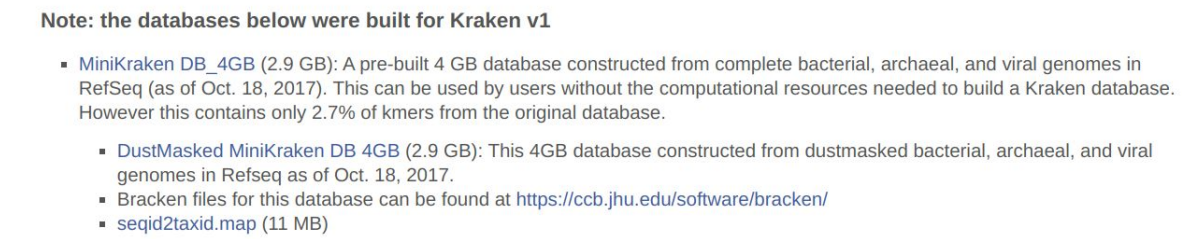
Раковый транскрипт:

	geneA.name	geneB.name	paircount	splitcount
43	BSG	NFIX	19	3
100	NPEPPS	TBC1D3D	6	1
107	NPEPPS	TBC1D3K	6	1
106	NPEPPS	TBC1D3L	6	1
105	NPEPPS	TBC1D3C	6	1
104	NPEPPS	TBC1D3H	6	1
103	NPEPPS	TBC1D3E	6	1
101	NPEPPS	TBC1D3	6	1
102	NPEPPS	TBC1D3G	6	1
99	NPEPPS	TBC1D3I	6	1
113	NOTCH1	NUP214	5	4
132	PTMA	HMGB1	5	2
56	EEF1A1	HSP90AB1	3	1
64	SET	PTMA	3	1
156	HSP90AB1	HSP90AA1	3	2
92	PPP1R12A	SEPTIN10	2	4
78	RPS2	FTH1	1	1
75	CENPX	DUS1L	1	1
74	RCN1	ERBB2	1	1
130	RPL15	CANX	1	1

Логично, что в раковом больше фьюжн-транскриптов, так как там по идее происходит больше мутаций всяких

6. Find viral transcripts from BLAST annotation/via kraken2 (https://github.com/DerrickWood/kraken2/wiki/Manual)

Кракен сам установлен первой версии, чтобы я мог использовать предсобранную БД:



Note: the databases below were built for Kraken v1

- MiniKraken DB_4GB (2.9 GB): A pre-built 4 GB database constructed from complete bacterial, archaeal, and viral genomes in RefSeq (as of Oct. 18, 2017). This can be used by users without the computational resources needed to build a Kraken database. However this contains only 2.7% of kmers from the original database.
  - DustMasked MiniKraken DB 4GB (2.9 GB): This 4GB database constructed from dustmasked bacterial, archaeal, and viral genomes in Refseq as of Oct. 18, 2017.
  - Bracken files for this database can be found at https://ccb.jhu.edu/software/bracken/
  - seqid2taxid.map (11 MB)

Usage:

```
kraken --db minikraken_20171013_4GB \
--threads 8 \
--paired \
--fastq-input \
data/SRR064287_1.fastq\
 data/SRR064287_2.fastq \
--only-classified-output \
--classified-out kraken_classified_cancer.fasta \
--output kraken_cancer.kraken
```

Out:

```
75748 sequences classified (1.11%)
  6724418 sequences unclassified (98.89%)
```

Make labels:

```
kraken-translate --db minikraken_20171013_4GB \
kraken_cancer.kraken > kraken_cancer.labels
```

And one more for non-cancer (normal):

Usage:

```
kraken --db minikraken_20171013_4GB \
--threads 8 \
--paired \
--fastq-input \
data/SRR064437_1.fastq\
 data/SRR064437_2.fastq \
--only-classified-output \
--classified-out kraken_classified_normal.fasta \
--output kraken_normal.kraken
```

Out:

```
40169 sequences classified (0.61%)
  6524388 sequences unclassified (99.39%)
```

Make labels:

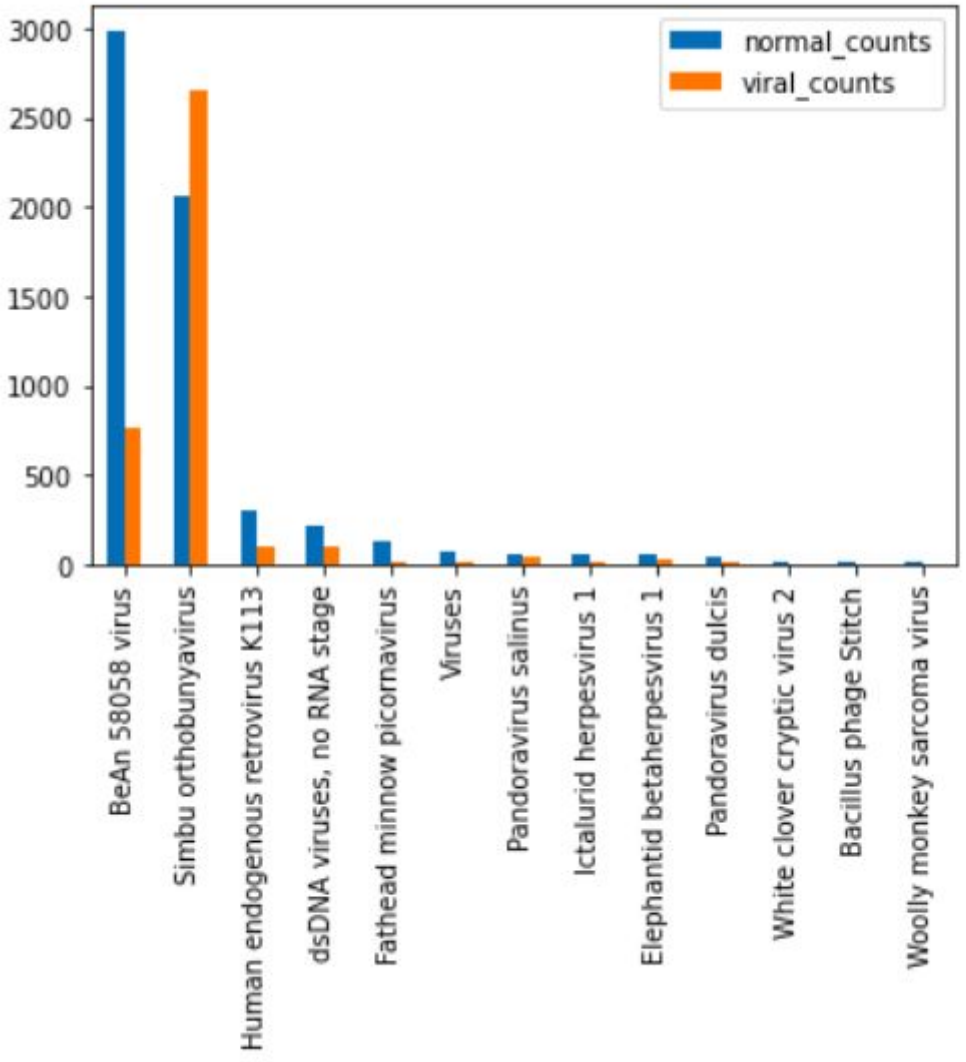
```
kraken-translate --db minikraken_20171013_4GB \
kraken_normal.kraken > kraken_normal.labels
```

Отфильтруем по вирусам:

```
(base) toharhymes@toharhymes-ThinkPad-L380:~/work/IB/2_gene_expression/homeworks/hw_3/all_data$ cat kraken_cancer.labels | grep -c Viruses
3788
(base) toharhymes@toharhymes-ThinkPad-L380:~/work/IB/2_gene_expression/homeworks/hw_3/all_data$ cat kraken_cancer.labels | grep -e Viruses > kraken_cancer_viruses.labels
(base) toharhymes@toharhymes-ThinkPad-L380:~/work/IB/2_gene_expression/homeworks/hw_3/all_data$ cat kraken_normal.labels | grep -c Viruses
6148
(base) toharhymes@toharhymes-ThinkPad-L380:~/work/IB/2_gene_expression/homeworks/hw_3/all_data$ cat kraken_normal.labels | grep -e Viruses > kraken_normal_viruses.labels
```

В нашем случае, получилось, что найдено больше в нормальных клетках (в 2 раза) - учитывая, что всего было 70 тысяч классифицировано в раковых, и 40 тысяч в не раковых.

Далее снова маленький анализ в питоне:
(Смержил, и отфильтровал, чтобы встречались хотя бы у 1 из сэмпла более 10 раз - получил график встречаемости рида каждого вируса)



В целом, можно сказать, что из всех ридов, очень маленький процент оказывается вирусным (в основном, все идентифицированные риды -

бактериальные). Причем, выделяется только два: BeAn 58058 Virus и Simbu orthobunyavirus. Интересные вирусы, один - родом из Бразилии, другой - из Уганды:)