Введение в анализ Ribo-Seq данных. Отчет

Горохов Никита

Ноябрь 2019

Этап 0

Моя рабочая директория: */mnt/local/vse2019/home/nsgorokhov/HW5/* Я установил миниконду и необходимые пакеты (svist4get, cutadapt, imagemagick). И проверил корректность их работы.

Этап 1

Идентификатор серии GEO: **GSE101865**. Теперь скачаем метаданные. Для этого воспользуемся командами:

```
$ ~/../kulakovskiy/edirect/esearch -db gds -query "GSE101865[ACCN]" | \
~/../kulakovskiy/edirect/efetch -format docsum -mode json > \
gse_details.json
$ ~/../kulakovskiy/edirect/esearch -db sra -query PRJNA395723 | \
~/../kulakovskiy/edirect/efetch -format runinfo > srx2srr.csv
```

Склеим два полученных файла

```
$ ruby ~/../kulakovskiy/h/h0_download_prep.rb gse_details.json \
srx2srr.csv > samples.tsv
```

И добавим в полученную табличку колонку с человеческими названиями экспериментов. В итоге финальная таблица будет иметь следующий вид: Для дальнейшего исследования я выбрал 4 файла (см. табл. 1). Теперь скачаем эти 4 файла (из-за того что все это происходит довольно долго, 2 файла я скачал и преобразовал в fasta, а два других взял из директории /../kulakovskiy/fastqraw/):

```
(base) nsgorokhov@hpc09:~/HW5$ cat samples_annotated.tsv
GSM2717371      SRX3033801      RNA_seq_Torin   SRR5865805      rna_seq_torin
GSM2717370      SRX3033800      RNA_seq_shABCF1_rep2    SRR5865804      rna_seq_shABCF1_rep2
GSM2717369      SRX3033799      RNA_seq_shABCF1_rep1    SRR5865803      rna_seq_shABCF1_rep1
GSM2717368      SRX3033798      RNA_seq_shMETTL3_rep2   SRR5865802      rna_seq_shMETTL3_rep2
GSM2717367      SRX3033797      RNA_seq_shMETTL3_rep1   SRR5865801      rna_seq_shMETTL3_rep1
GSM2717366      SRX3033796      RNA_seq_Scram_rep2      SRR5865800      rna_seq_Scram_rep2
GSM2717365      SRX3033795      RNA_seq_Scram_rep1      SRR5865799      rna_seq_Scram_rep1
GSM2717364      SRX3033794      Ribo_seq_Torin  SRR5865798      ribo_seq_torin
GSM2717363      SRX3033793      Ribo_seq_shABCF1_rep2   SRR5865797      ribo_seq_shABCF1_rep2
GSM2717362      SRX3033792      Ribo_seq_shABCF1_rep1   SRR5865796      ribo_seq_shABCF1_rep1
GSM2717361      SRX3033791      Ribo_seq_shMETTL3_rep2  SRR5865795      ribo_seq_shMETT3_rep2
GSM2717360      SRX3033790      Ribo_seq_shMETTL3_rep1  SRR5865794      ribo_seq_shMETT3_rep1
GSM2717359      SRX3033789      Ribo_seq_Scram_rep2     SRR5865793      ribo_seq_Scram_rep2
GSM2717358      SRX3033787      Ribo_seq_Scram_rep1     SRR5865792      ribo_seq_Scram_rep1
```

Рис. 1: Таблица с метаданными

id	описание	адаптер
SRR5865803	RNA Seq ABCF1 replicate 1	AAAAAAAAAA
SRR5865801	RNA Seq METTL3 replicate 1	AAAAAAAAAA
SRR5865796	RIBO Seq ABCF1 replicate 1	AAAAAAAAAA
SRR5865794	Ribo Seq METTl3 replicate 1	AAAAAAAAAA

Таблица 1: Информация о выбранных данных

```
$ ~/../kulakovskiy/bin/sratoolkit/sratoolkit.2.10.0-ubuntu64/bin/prefetch
-v SRR5865803
```

И распакуем в директорию fasta

```
$ ~/../kulakovskiy/bin/sratoolkit/sratoolkit.2.10.0-ubuntu64/bin/fastq-du
--gzip SRR5865803 --split-files -O ../../../HW5/fasta/
```

Также создадим сивольные ссылки на эти fasta файлы (они будут храниться в каталоге fasta_ln/)

```
$ ln fasta/SRR5865803_1.fasta.gz fasta_ln/rna_seq_shABCF1_rep1
```

Этап 2

Запустим FastQC для 4 образцов.

```
$ ~/../kulakovskiy/bin/fastqc/FastQC/fastqc fasta/SRR5865803_1. \
fastq.gz -o fastqc_before_trim/
```

После этого удалим адаптеры и повторно запустим FastQC для обновленных fastq файлов (для результатов Ribo seq я дополнительно выставлял паратмер –trimmed-only).

```
$ cutadapt -a AAAAAAAAAA -j 4 --minimum-length 20 \
-q 20 fasta/SRR5865803_1.fasrq.gz -o \
fasta_trim/SRR5865803_after_trim.fastq.gz
```

Результаты работы FastQC до и после тримминга находятся в папках fastqc_before_trim и fastqc_after_trim соответственно. Для упрощения, будем работать только с одной парой данных (ABCF1 rna и ribo seq) Теперь проанализируем полученые результаты

1. **Вопрос**: Какова характерная длина рибосомных футпринтов в рибосеке и рнксеке? **Ответ** Для рнк и рибо экспериментов длина футпринтов равна 49 (поле Sequence length в Basic Statistics) до тримминга и 39 (рнасек) и 32 (рибосек).

2. **Вопрос**: Обрабатывали ли РНК-сек по тому же протоколу (с нарезкой нуклеазами и отбором фрагментов по размеру) или нет. **Ответ** Нет.

Этап 3

Проверим долю рибосомной РНК.

```
$ ~/../kulakovskiy/bin/bowtie-1.2.3-linux-x86_64/bowtie --sam -p 4 \
~/../kulakovskiy/bin/bowtie-1.2.3-linux-x86_64/rRNA_euk/rRNA_euk \
SRR5865803_after_trim.fastq.gz --chunkmbs 10000 > /dev/null
```

Итого для рибосека (46.15%):

- reads processed: 53393979
- reads with at least one reported alignment: 24639869 (46.15%)
- reads that failed to align: 28754110 (53.85%)

Итого для рнксека (49.18%):

- reads processed: 62791081
- reads with at least one reported alignment: 30881163 (49.18%)
- reads that failed to align: 31909918 (50.82%)

Этап 4

Этап 5

Фазирование прочтений и метагенные профили. Для выполнения этой задачи нам понадобится пакет **plastid**. Геномные аннотации мы будем брать из */../kulakovskiy/genomes/plastidme* Команда запуска имеет следующий вид (для рибосека):

```
$ psite ~/../kulakovskiy/genomes/plastidmetagen/mouse_start_rois.txt

psite_test --countfile_format BAM --count_files \

bam/ABCF1_ribo_Coots2017_m_r1.bam \

--min_length 20 --max_length 40 --aggregate --constrain 10 18 \

--min_count 10 --default 14
```

Аналогично запустим для результатов рнасека. В результате получаем файл с фазированием прочтений различной длины и картинку фазирования. На рис.2 изображено фазирование для рибо и рнксека. Можно сделать вывод, что для рнксека фазирование не получилось.

Теперь перейдем к построению метогеномного профиля. Для этого воспользуемся командой **metagene**:

```
$ metagene count --countfile_format BAM --count_files \

bam/ABCF1_ribo_Coots2017_m_r1.bam \

--fiveprime --min_length 25 --max_length 32 --min_count 10 \

--use_mean --landmark Start \

~/../kulakovskiy/genomes/plastidmetagen/mouse_start_rois.txt metagene_cou
```

Аналогично сделаем для рнксека и сравним полученные профили. На рис.3 изображены профили для рибо и рнксека.

Этап 6

Получение bedGraph файлов и визуализация. Воспользуемся утилитой **make_wiggle**.

```
$ make_wiggle -o output --count_files

ABCF1_ribo_Coots2017_m_r1.bam --normalize

--min_length 25 --max_length 31 --fiveprime_variable

--offset psite_test_p_offsets.txt
```

На выходе получаем 2 файла: **output_fw, output_rc**. Далее склеим эти два файла в один. Для это установим пакет **csvtk** (conda install -c bioconda csvtk).

```
$ bedtools unionbedg -i output_fw.wig output_rc.wig | \

csvtk mutate2 -H -t -L 5 -e '$4+$5' | \

cut -f 4-5 --complement > output_ribo.bedGraph
```

Аналогично поступаем с рнксек файлом. Теперь с помощью **svist4get** провизулизируем некоторые гены. Я выбрал *TP53* (ENSMUSG00000059552) и *ABCF1* (ENSMUSG00000038762). В результате получил следующие картинки:

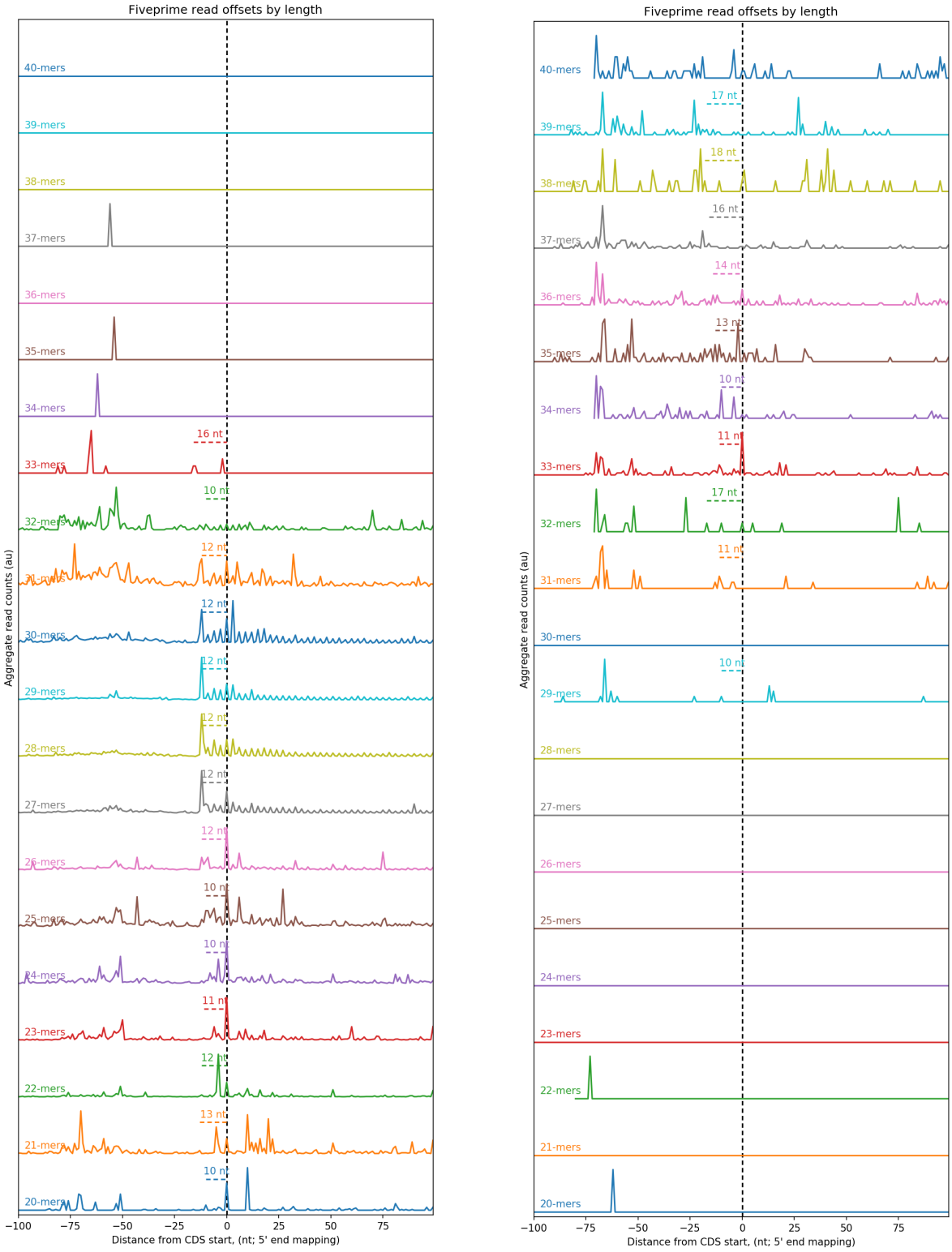

Рис. 2: Рибосек (слева) и рнксек (справа)

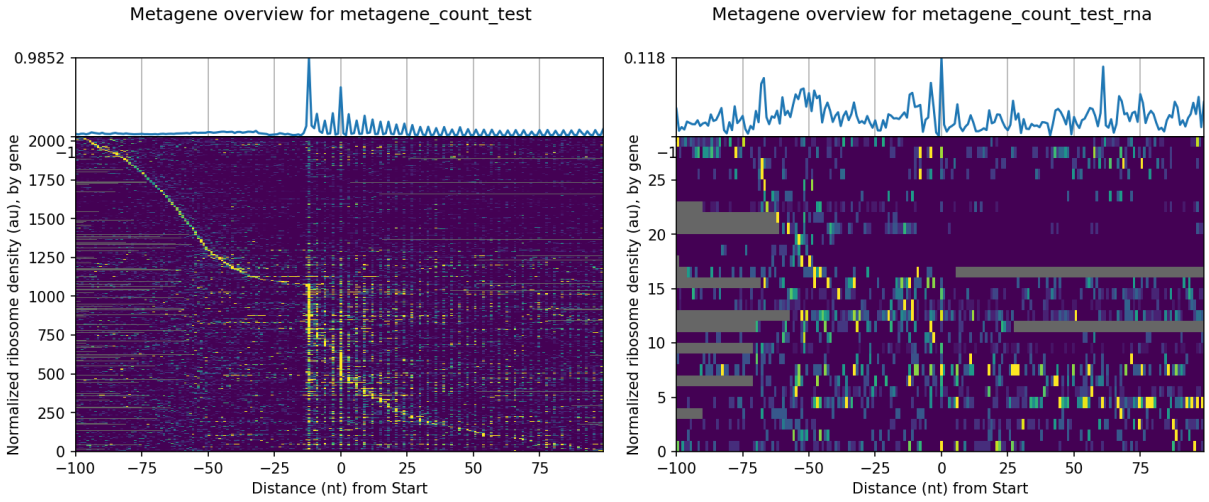

Рис. 3: Рибосек (слева) и рнксек (справа)

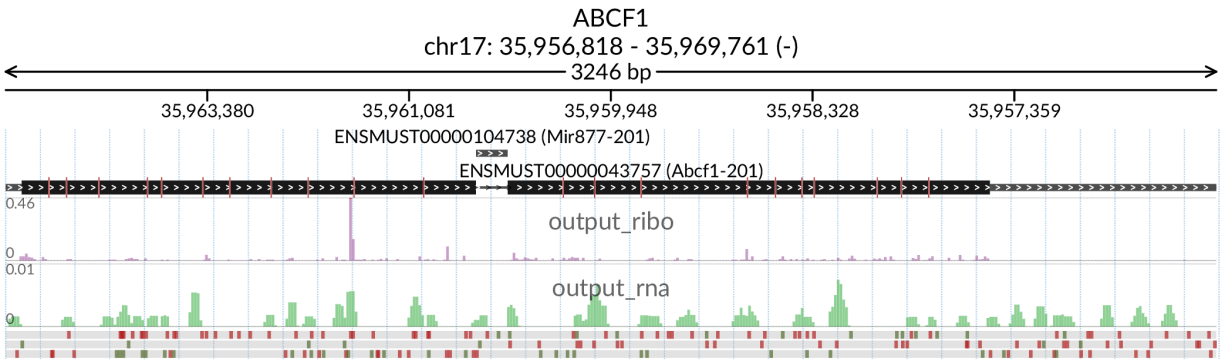

Рис. 4: ген ABCF1

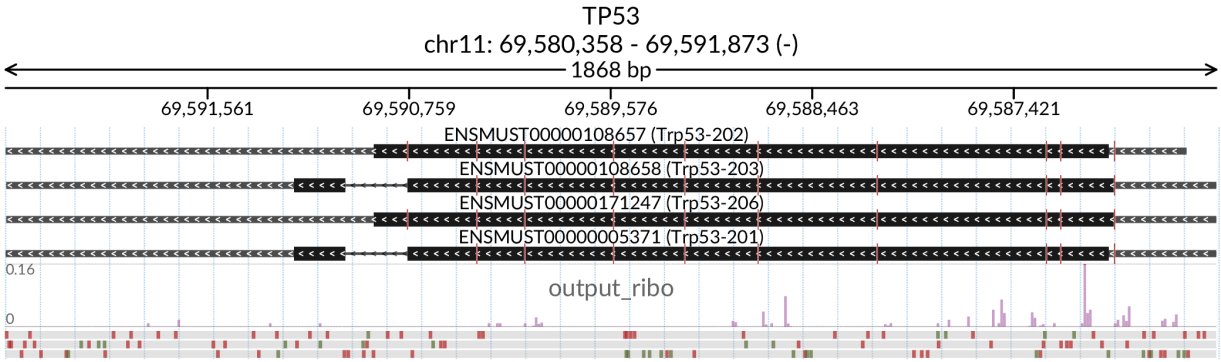

Рис. 5: ген TP53