

Для того, чтобы сдать задания практикумов 11-12, необходимо подготовить программный сценарий, который:

- на вход принимает ID файла и имя хромосомы
- выдает фильтрованный vcf файл

Подготовьте программный сценарий (и конфигурационный файл, если решите использовать такую организацию).

На коллоквиуме нужно будет показать скрипт и ответить на вопросы о любых параметрах, типе входных и выходных файлов, для чего нужна та или иная строка, программа, параметр и т.д.

Не удаляйте сам скрипт, а также входные и выходные файлы, получаемые в ходе исполнения программного сценария. На коллоквиуме я могу попросить показать мне любой файл, который генерит ваш программный сценарий. Ваш скрипт должен быть в рабочем состоянии, т.е. я могу вам дать тестовые входные файлы и ваш скрипт должен с этим справиться.

Примеры оформления кода

1. Базовый вариант

Каждая команда записывается ровно так, как она и должна быть записана для выполнения, с абсолютными или относительными путями.

```
====script.sh====

#!/bin/bash

fastqc /mnt/scratch/NGS/exome/reads/SRR001.fastq.gz -o /mnt/scratch/NGS/exome/fastqc

TrimmomaticSE -phred33 /mnt/scratch/NGS/exome/reads/SRR001.fastq.gz
/mnt/scratch/NGS/exome/reads_trim/SRR001.fastq.gz TRAILING:3 MINLEN:36
```

Запуск: ./script.sh

Здесь и далее – убедитесь, что у вашего скрипта есть права на исполнение. Такой вариант годится для зачета, но помните про тестовые входные файлы.

2. Выносим заменяемые части в переменные

```
====script.sh====

#!/bin/bash

sample=SRR001
raw_reads=/mnt/scratch/NGS/exome/reads
fastqc=/mnt/scratch/NGS/exome/fastqc
trim_reads=/mnt/scratch/NGS/exome/reads_trim
trail=3
minlen=36

fastqc $raw_reads/$sample.fastq.gz -o $fastqc

TrimmomaticSE -phred33 $raw_reads/$sample.fastq.gz $trim_reads /$sample.fastq.gz TRAILING:$trail
MINLEN:$minlen
```

Запуск: ./script.sh

Все пути, значения параметров разных программ и прочие части скрипта, которые могут быть изменены при других входных данных, при использовании вашего скрипта на другом кластере и т.п. могут быть вынесены в отдельный блок с переменными.

3. Выносим имя образца как параметр при запуске скрипта

```
=====script.sh=====

#!/bin/bash

sample=$1
raw_reads=/mnt/scratch/NGS/exome/reads
fastqc=/mnt/scratch/NGS/exome/fastqc
trim_reads=/mnt/scratch/NGS/exome/reads_trim
trail=3
minlen=36

fastqc $raw_reads/$sample.fastq.gz -o $fastqc

TrimmomaticSE -phred33 $raw_reads/$sample.fastq.gz $ trim_reads /$sample.fastq.gz TRAILING:$trail
MINLEN:$minlen
```

Запуск: ./script.sh SRR001

В этом варианте имя образца задано как первый параметр при запуске скрипта.
Также изменяемые части скрипта вынесены в переменные в начало.

4. Выносим все переменные в конфигурационный файл

```
====config.cfg====
```

```
sample=$1
raw_reads=/mnt/scratch/NGS/exome/reads
fastqc=/mnt/scratch/NGS/exome/fastqc
trim_reads=/mnt/scratch/NGS/exome/reads_trim
trail=3
minlen=36
```

```
====script.sh====
```

```
#!/bin/bash

cfg=$2
. $cfg # "чтение" конфигурационного файла!!!

fastqc $raw_reads/$sample.fastq.gz -o $fastqc

TrimmomaticSE -phred33 $raw_reads/$sample.fastq.gz $ trim_reads /$sample.fastq.gz TRAILING:$trail
MINLEN:$minlen
```

Запуск: ./script.sh SRR001 ./config.cfg

В этом варианте вы создаете отдельный файл, куда выносятся ВСЕ возможные изменяемые части скрипта. Все пути, все параметры, все имена.

При таком подходе вы никогда не залезаете в тело рабочего скрипта и ничего в нем не меняете. Конфигурационный файл служит для вас своего рода лабораторным журналом, потому что вы можете сохранять каждый раз этот файл за нужной датой и менять в нем что угодно, гарантируя техническое воспроизведение вашего анализа.