Instrukcja do ćwiczenia nr 3 z MBI
Resekwencjonowanie genomu człowieka

Tomasz Gambin, Wiktor Kuśmirek, Robert Nowak

2024Z

1 Wstęp

Analiza danych z resekwencjonowania genomu o znanej sekwencji referencyjnej (jak w przypadku genomu człowieka) składa się z następującth etapów:

- mapowaniu i uliniowieniu sekewncji odczytów do genomu referencyjnego (FASTQ –> BAM)
- wykrywaniu wariantów
- adnotacji wariantów
- filtrowaniu, priorytetyzacji i interpretacji wariantów

2 Instalacja narzędzi i pobranie danych wejściowych

Instrukcja pokazuje poszczególne kroki na Ubuntu 20.04 w wersji 64 bitowej. Wymagania wstępne to instalacja aplikacji docker.

`https://docs.docker.com/engine/install/ubuntu/`

Pobierz obrazy dockerowe dla narzędzi bwa, samtools, bcftools:

```
sudo docker pull quay.io/biocontainers/bwa:0.7.17--hed695b0_7
sudo docker pull biocontainers/samtools:v1.9-4-deb_cv1
sudo docker pull biocontainers/bcftools:v1.9-1-deb_cv1
```

Zainstaluj program IGV:

```
sudo apt-get install igv
```

Utwórz katalog roboczy oraz pobierz dane:

```
mkdir ~/mbi_cwiczenie3/
cd ~/mbi_cwiczenie3/
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/chr1.fa.gz
gunzip chr1.fa.gz
```

Pobierz plik coriell_chr1.fq.gz:

`https://drive.google.com/file/d/1UU2IlgQ58TerqkZglASchab5Iu9_kSKQ/view?usp=sharing`

Rozpakuj pobrany plik:

```
gunzip coriell_chr1.fq.gz
```

i umieść w folderze mbi_cwiczenie3 Po tej operacji plik chr1.fa powinien mieć wielkość ok. 243MB, zaś coriell_chr1.fq ok. 55MB.

3 Mapowanie

Zapoznaj się z opcjami narzędzia bwa:

```
sudo docker run -v ~/mbi_cwiczenie3:/data quay.io/biocontainers/bwa:0.7.17--hed695b0_7 \
  bwa
```

Zaindeksuj plik fasta genomu referencyjnego (ok. 5 min):

```
sudo docker run -v ~/mbi_cwiczenie3:/data quay.io/biocontainers/bwa:0.7.17--hed695b0_7 \
  bwa index /data/chr1.fa
```

Zapoznaj się z opcjami algorytmu bwa mem:

```
sudo docker run -v ~/mbi_cwiczenie3:/data quay.io/biocontainers/bwa:0.7.17--hed695b0_7 \
  bwa mem
```

Przeprowadź mapowanie za pomocą algorytmu bwa mem i wygeneruj plik SAM:

```
sudo docker run -v ~/mbi_cwiczenie3:/data quay.io/biocontainers/bwa:0.7.17--hed695b0_7 \
  bwa mem -t 4 /data/chr1.fa /data/coriell_chr1.fq -o /data/coriell_chr1.sam
```

Sprawdź zawartość wygenerowango pliku SAM. Jaka jest typowa długość odczytów?

Dokonaj sortowania i wygeneruj plik BAM:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/samtools:v1.9-4-deb_cv1 \
  samtools sort -O BAM -o coriell_chr1.bam coriell_chr1.sam
```

Jak jest różnica w wielkości plików FASTQ, BAM, SAM?

4 Wizualizacja zawartości pliku BAM w programie IGV

Przygotuj indeks dla pliku BAM (konieczny do wczytania BAM w programie IGV).

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/samtools:v1.9-4-deb_cv1 \
  samtools index coriell_chr1.bam
```

Uruchom program IGV (np. wywołaj w bash "igv"). Instrukcja do programu znajduje się na stronie `http://software.broadinstitute.org/software/igv/`.

Wybierz wersję genomu ludzkiego (hg19) i załaduj wygenerowany plik BAM (File $->$ Load from file).

Wyszukaj w IGV gen IQGAP3 (wpisz nazwę genu w oknie wyszkuiwania).

Wyszukaj jeden wariant o pokryciu całkowitym powyżej 10x. **Jaka jest pozycja tego wariantu? Ile odczytów wskazuje na wariant a ile na referencje? Czy jest to wariant homo czy heterozygotyczny? Załącz zrzut ekranu z programu IGV pokazujący wybrany wariant.**

5 Wykrywanie wariantów

W niniejszym ćwiczeniu, wykrywanie wariantów zostanie przeprowadzone przy wykorzystaniu narzędzi samtools i bcftools. W pierwszym kroku dokonamy wyznaczenia tzw. pileup'ów zapisywanych do pliku BCF. W drugim program bcftools zostanie wykorzystany do właściwej identyfikacji listy wariantów, która zostanie zapisana do pliku VCF.

Zapoznaj się z możliwościami progamów samtools:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/samtools:v1.9-4-deb_cv1 \
  samtools
```

i bcftools:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/bcftools:v1.9-1-deb_cv1 \
  bcftools
```

W szczególności przeczytaj instrukcję dla komendy samtools mpileup:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/samtools:v1.9-4-deb_cv1 \
  samtools mpileup
```

oraz bcftools call:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/bcftools:v1.9-1-deb_cv1 \
  bcftools call
```

Wygeneruj plik BCF i przeprowadź identyfikację wariantów:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/samtools:v1.9-4-deb_cv1 \
  samtools mpileup -Ou -f chr1.fa coriell_chr1.bam > coriell_chr1.bcf

sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/bcftools:v1.9-1-deb_cv1 \
  bcftools call -mv  coriell_chr1.bcf  > coriell_chr1.vcf
```

Ile wariantów zawiera plik?

Usuń warianty z całkowitym pokyciem poniżej 11x:

```
sudo docker run -v ~/mbi_cwiczenie3:/data biocontainers/bcftools:v1.9-1-deb_cv1 \
  bcftools filter -i "INFO/DP> 10" coriell_chr1.vcf > coriell_chr1_filtered.vcf
```

Ile wariantów zostało po filtracji? Jakich innych parametrów możemy użyć do dalszej filtracji liczby wariantów?

6 Adnotacje wariantów

Przeprowadź adnotację przefiltrowanego pliku VCF wykorzystując narzędzie VEP w wersji online:

`http://grch37.ensembl.org/Homo_sapiens/Tools/VEP`

Załaduj plik "$coriell\_chr1\_filtered.vcf$" w polu "or upload file Choose File", a następnie naciśnij przycisk "Run" i poczekaj na zakończenie obliczeń. Po zakończeniu, kliknij przycisk "View results".

Przejrzyj statystki wariantów zaprezentowane na pierwszym wykresie typu "pie chart". **Jaki typ wariantu przeważa?**

Pobierz zaadnotowaną listę wariantów w formacie txt (Download TXT). Wyszukaj wariant, który wcześniej zidentyfikowałeś ręcznie w programie IGV. **Załącz do sprawozdania wiersze odpowiadające temu wariantowi. Czy jest to wariant w części kodującej?**

7 Zadanie implementacyjne

- Proszę zapoznać się z formatem pliku refFlat:
  `https://genome.ucsc.edu/goldenPath/gbdDescriptions.html`,
  oraz pobrać jego zawartość z:
  `https://hgdownload.cse.ucsc.edu/goldenpath/hg19/database/refFlat.txt.gz`

- Proszę napisać skrypt, który wyliczy ile wariantów (z pliku $coriell\_chr1.vcf$) znajduje się w poszczególnych genach, których współrzędne znajdują się w pliku refFlat. Jako początek i koniec genu należy przyjąć kolumny txStart i txEnd.

  Skrypt powinien zwracać tabelę z dwiema kolumnami (symbol genu, liczba wariantów). Należy dokonać implementacji w języku R z wykorzystaniem pakietu GenomicRanges `https://bioconductor.org/packages/release/bioc/html/GenomicRanges.html` lub w języku python z wykorzystaniem bibblioteki pyranges `https://github.com/biocore-ntnu/pyranges`.