Nanopore data pre-processing

Seong-Kun Bak (sanekun@gmail.com)

2022 7 11

Introduce

basecalling 이 끝난 후 raw data는 목적(read length, Q-score ..)에 맞는 pre-processing 과정을 거친다. 해당 과정을 통해 과도한 데이터 크기를 줄이고 원하는 범위의 데이터 혹은 높은 수준의 데이터만을 활용 가능하다.

많이 사용하는 illumina 시스템의 경우 여기에서 기술된 과정보다 훨씬 많은 과정이 포함된다. 이는 애초에 데이터 양이 매우 많아 (>100GB) 훨씬 포괄적인 범위로 필터링을 진행하여도 충분한 데이터를 확보할 수 있고, duplicated, chimera read가 많아 제거가 필요하기 때문이다.

그러나, 소개한 nanopore long-read sequencing은 데이터에 불필요한 부분(adapter, barcode..)이 상대적으로 매우 적고 전체 데이터 양(throughput)도 적어 전처리 과정이 필수적이지는 않다. 그럼에도 데이터 분석의 편의성, 약간의 quality 상승을 위해 전처리를 해주는 것이 좋다.

아래 `method`를 통해 전체 과정을 보여주고 `subsystem`에서 개별적인 프로그램의 설명을 기술하겠다.

Programs

NanoFilt : Read의 평균 q-score, length를 기준으로 filtering

- 기본적으로 Q7, Q10등 너무 낮은 quality는 삭제하는 것이 좋음.
- 여러 길이의 DNA가 존재할 때 raw data의 크기를 크게 줄여 줌
- Assembly DNA등 길이 를 활용한 Filtering 만으로도 분석에 충분한 정보를 획득 할 수 있음.
  [Figure]

porechop : ONT read 양 말단에 존재하는 barcode + adapter 서열 (약 40 bp)을 trimming 해줌.

- 필수는 아니나 사용중인 barcode (tag)가 기존 ONT barcode와 겹친다면 제거하는 것이 좋음 (ONT barcode 서열은 nanopore protocol 내 표기되어있음)

minimap2 : long-read 데이터에 최적화된 assembler, error-rate가 높은 환경에서 assembly가 가능 하도록 설계됨.

- 분석 방향에 맞추어 score matrix 변경 가능
- circular mapping이 안되어 아래 `circulaization` 으로 해결하였음.

samtools : alignment format (sam, bam)을 다루기 위한 기본적인 프로그램

pysam : samtools와 비슷하며 alignment format을 python으로 pipe 하기위한 패키지

- python으로 작성한 대부분의 프로그램에 사용 하였으며 개인 마다 최적화된 분석을 하기에 용이하다.

Personal Programs

circular_transformation : read들의 strand, 위치를 고정하기 위하여 제작됨.

- Transposon 기반의 tagmentation시 read의 시작 위치가 서로 다르기 때문에 mapping의 품질이 떨어진다. (minimap2 는 circular mapping을 지원하지 않음)
- CDS와 같이 특정 영역 사이에 barcode가 들어가 결과가 왜곡 되는 것을 방지하고 pysam을 활용한 개별 분석을 편하게 해줌.
  = 따라서 모든 read의 시작위치를 고정할 시 더 좋은 결과를 얻을 수 있음.

Bar_parser : pysam을 사용하여 특정 parameter들을 piping 하기 위한 프로그램

Usage

NanoFilt

Porechop

circular_transformation