Assembly Megahit

2022-01-20

20193852 문유빈

1) Goal : Megahit 프로그램을 사용해 assembly 하기

2) Shotgun pipeline(*현재 진행된)
Raw data -> QC -> FastUniq -> (Host 제거) -> Assembly

* 현재 하고 있는 샘플은 물이므로 따로 host 제거 과정이 필요하지 않음
* Assembly는 짧은 reads를 이어 contig로 만드는 과정

3) Megahit

- MEGAHIT is an ultra-fast and memory-efficient NGS assembler. It is optimized for metagenomes, but also works well on generic single genome assembly (small or mammalian size) and single-cell assembly.

- 빠르고 괜찮은 퀄리티를 자랑
** Assembly tool들은 각각의 장단점이 존재
-> assembly comparison 검색 시 다양한 툴 비교 가능

- Parameter
◎ -1 <pe1> comma-separated list of fasta/q paired-end #1 files, paired with files in <pe2> → 첫 번째 파일, 파일2와 pair
◎ -2 <pe2> comma-separated list of fasta/q paired-end #2 files, paired with files in <pe1> → 두 번째 파일, 파일1과 pair
◎ -r <se> comma-separated list of fasta → unpaired 파일 1개

◎ -presets <str>override a group of parameters;
possible values:
○ meta-sensitive: '--min-count 1 –k-list 21,29,39,49,...,129,141'
→ 리드를 21, 29, 39, 49. ... , 141로 자름
○ meta-large: '--k-min 27 --k-max 127 –k-step 10'(large & complex metagenomes, like soil) → 리드를 27, 37, 47, ... , 127로 자름

** 토양 같은 파일(대용량)은 meta-large 사용, 우린 meta-sensitive 사용

** k-mer

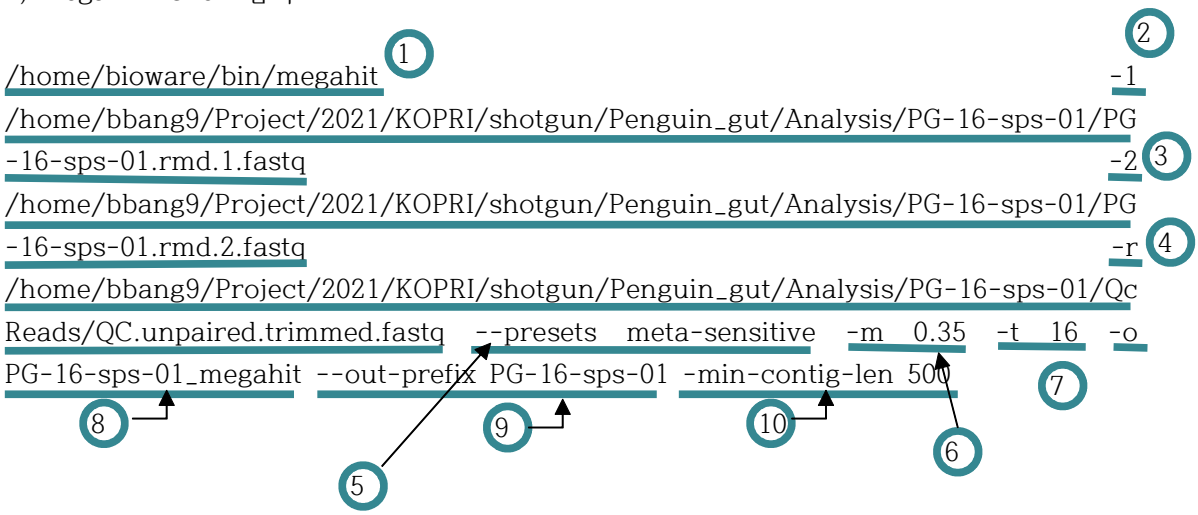
k-mer는 일반적으로 문자열(string)에서 가능한 모든 부분문자열(substring)의 길이 k를 의미한다. 유전체학에서 k-mer는 DNA sequencing으로 얻은 read의 모든 가능한 부분 서열(의 길이 k)을 의미한다. L이라는 길이의 문자열이 주어졌을 때, k-mer의 양은 L - k + 1 이며, n 개(예를 들어 DNA의 ATCG의 경우 4)에 대해 가능한 k-mer의 개수는 nk 이다.

k-mer는 일반적으로 sequence assembly에 사용되지만, sequence alignment에도 사용될 수 있다.인간 유전체 관점에서, 돌연변이율의 가변성을 설명하는데 다양한 길이의 k-mer가 사용되었다.
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=naturelove87&logNo=221561868710

◎ -m <float> max memory in byte to be used in SdBG construction
◎ -t <int> number of CPU threads, at least 2 if GPU enabled.

◎ -o <string> output directory [./megahit_out]
◎ --out-prefix <string> output prefix → 생성될 파일 이름 설정
◎ --min-contig-len <int> minimum length of contigs to output
→ default = 200, 우린 500으로 설정해 최소 500bp 이상만

4) Megahit xshell 입력



xshell 입력 코드

```
[guest01@smel0:yoFastUniq]$ /home/bioware/bin/megahit -1 /home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/PG-16-sps-01.rmd.1.fastq -2 /home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/PG-16-sps-01.rmd.2.fastq -r /home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/QcReads/QC.unpaired.trimmed.fastq --presets meta-sensitive -m 0.35 -t 16 -o PG-16-sps-01_megahit --out-prefix PG-16-sps-01 --min-contig-len 500
```

① /home/bioware/bin/megahit => megahit 절대 경로

* 절대 경로 모를 때는 xshell에 which megahit 입력하면 나옴

```
[guest01@smel0:PG-16-sps-01_megahit]$ which megahit
/home/bioware/bin/megahit
```

② -1 첫 번째 파일 경로/파일 명
③ -2 두 번째 파일 경로/파일 명
④ -r unpaired 파일 경로/파일 명

진주 언니 파일 경로

⑤ --presets meta-sensitive
⑥ -m 0.35
⑦ -t 16
⑧ -o PG-16-sps-01_megahit
⑨ --out-prefix PG-16-sps-01
⑩ --min-contig-len 500

내 경로에 생성되는 output 파일

5) 실행 경과

```
503.0Gb memory in total.
Using: 176.271Gb.
MEGAHIT v1.1.3
--- [Thu Jan 20 15:22:03 2022] Start assembly. Number of CPU threads 16 ---
--- [Thu Jan 20 15:22:03 2022] Available memory: 540770074624, used: 189269526118
--- [Thu Jan 20 15:22:03 2022] Converting reads to binaries ---
   [read_lib_functions-inl.h  : 209]     Lib 0 (/home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/PG-16-sps-01.rmd.1.fastq,/home/bbang9/Project/2021/KOPRI/shotgun
/Penguin_gut/Analysis/PG-16-sps-01/PG-16-sps-01.rmd.2.fastq): pe, 37757396 reads, 151 max length
   [read_lib_functions-inl.h  : 209]     Lib 1 (/home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/QcReads/QC.unpaired.trimmed.fastq): se, 317411 reads, 151 max le
ngth
   [utils.h                   : 126]     Real: 62.7013 user: 38.1018   sys: 4.1720     maxrss: 162800
--- [Thu Jan 20 15:23:06 2022] k list: 21,29,39,49,59,69,79,89,99,109,119,129,141 ---
--- [Thu Jan 20 15:23:06 2022] Extracting solid (k+1)-mers and building sdbg for k = 21 ---
--- [Thu Jan 20 15:29:18 2022] Assembling contigs from SdBG for k = 21 ---
--- [Thu Jan 20 16:04:12 2022] Local assembling k = 21 ---
--- [Thu Jan 20 16:08:53 2022] Extracting iterative edges from k = 21 to 29 ---
--- [Thu Jan 20 16:13:27 2022] Building graph for k = 29 ---
--- [Thu Jan 20 16:21:50 2022] Assembling contigs from SdBG for k = 29 ---
```

```
[guest01@smel0:PG-16-sps-01_megahit]$ ll
total 8
drwxrwxr-x 2 guest01 guest01   10 Jan 20 15:22 intermediate_contigs
-rw-rw-r-- 1 guest01 guest01  469 Jan 20 15:22 opts.txt
-rw-rw-r-- 1 guest01 guest01 3288 Jan 20 15:25 PG-16-sps-01.log
drwxrwxr-x 3 guest01 guest01  114 Jan 20 15:23 tmp
```

실행 중
몇 시간 소요됨

```
[guest01@smel0:PG-16-sps-01_megahit]$ ll
total 20
drwxrwxr-x 2 guest01 guest01 4096 Jan 20 15:48 intermediate_contigs
-rw-rw-r-- 1 guest01 guest01  469 Jan 20 15:22 opts.txt
-rw-rw-r-- 1 guest01 guest01 9524 Jan 20 15:48 PG-16-sps-01.log
drwxrwxr-x 3 guest01 guest01  114 Jan 20 15:23 tmp
```

6) 결과

>ybMegahit 폴더

```
[guest01@smel0:ybMegahit]$ ll
total 0
drwxrwxr-x 3 guest01 guest01 136 Jan 21 10:01 PG-16-sps-01_megahit
```

>PG-16-sps-01_megahit 폴더

```
[guest01@smel0:PG-16-sps-01_megahit]$ ll
total 771532
-rw-rw-r-- 1 guest01 guest01         0 Jan 20 20:18 done
drwxrwxr-x 2 guest01 guest01      4096 Jan 21 10:04 intermediate_contigs
-rw-rw-r-- 1 guest01 guest01       469 Jan 20 15:22 opts.txt
-rw-rw-r-- 1 guest01 guest01 789891711 Jan 20 20:18 PG-16-sps-01.contigs.fa
-rw-rw-r-- 1 guest01 guest01    145715 Jan 20 20:18 PG-16-sps-01.log
```

* intermediate_contigs 폴더는 assembly 하면서 중간에 생긴, 사용한 것들
* opts.txt엔 내가 넣은 옵션

```
-1
/home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/PG-16-sps-01.rmd.1.fastq
-2
/home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/PG-16-sps-01.rmd.2.fastq
-r
/home/bbang9/Project/2021/KOPRI/shotgun/Penguin_gut/Analysis/PG-16-sps-01/QcReads/QC.unpaired.trimmed.fastq
--presets
meta-sensitive
-m
0.35
-t
16
-o
PG-16-sps-01_megahit
--out-prefix
PG-16-sps-01
--min-contig-len
500
MEGAHIT_TEMP_DIR: PG-16-sps-01_megahit/tmp/
```

*** PG-16-sps-01.contigs.fa => 우리가 쓸 파일

```
>k141_2 flag=1 multi=4.0000 len=515
AGAAGCAGGACAGGTAGAAAGACGCACAAACTAAAGCAATGTTTTTTGGCTGTGTTACATTACGTACATAAATTTACAGACATACATACACCAGTTAATCAATCATCTCCAGTCAGAAATGTCGGGGTATCTTAAATGATATTGTACCATGGGACTGTGTTCAGGCAATTCCTTTTCACTCTATTAT
AACTTTTCATCTTGAGCCAATATTAAATTCCATTTATCACAATAGGTGGCAGTAGCTATACCATTTTCAGCAGCATCTTCTAGATAAAAACACCCCAATGTCTCTCTGCAGCACCATTAGCAGTATTGACTGCTCTATTTTCTCTCAATAGAAAGCCCCTCACAGATACCTTATTGTGGAAAAGGCA
TCCTCAGCAGCACTAAACATCTTTTGAGCAACATTTGAAGGCGACGAGATTAAATACTCAGCAAGAGTTATGGCAAGGCTCTACGCCATTTTCATTTAGGGAATTTGATCATTATGGGGATGTGCTTTTCCTTGTTGGTGA
>k141_3 flag=1 multi=3.0000 len=648
AATTCTTTGCAGCTGCCCAGGCAGAAAGCCAGACCTTTCAGGTGGTGGGTTATGGGCAGAACAGAAGGAGTTCTGGCATCCTCACAAACAGGACACCCTTGGGCAGGGGAAAGCCCAGAAAATACCAAAGTGCCCAGGACTGCTCTCCCACATGGGCTTCGTGCCCAAGATAAGGAGCATCCAGCCA
GGTGGTAGAACATAGCGATGGGACTGCAAGAGCCAGGCCACGGGGAGGGAGAAGGGATGTGGCTCAACTCCACTGTCTTCACCTGGGACACACGGCTTTATGTGCCTTACATAGGAGCAGAAAGTAGGGACCTTTGATGCATGTGCACAGTAAAATTAGTGTAGAGGTTAAAAAACTCTTTTGCCCC
ATAGTCCTGCCATTCTACGTCATCCCTCCATCTTCCCTCTGTCATGCTGACTAATCCTCTGCAGTTTGTGTTCTCCTCTCCGTAATTTTATTTTTTCTGTTTGAAAATAGTTGGAACCCTTGTGATTTCTTTGCTTTATTGTTATATTTCTCTTCAAACAAATGCTCTCTAATAGAGCCAGTCTTTG
ACACAGCACCTTCCCTTCCCCTAACAGATTCTGTCTCCCTATGGGACAGACCCTCTACTTTCACACATCTGCTTGGGTGCTGGAGCT
>k141_4 flag=1 multi=5.0000 len=512
CCTACACGACGCTCTTCCGATCTCTGTTATGGGAGGAAACCTGAATGGAGCACCAGAGCTCCCTCAAATTCCTCAGCAAAACTATGGAGAACTTCAACTCAGTGGTGCAGAAGCAGAGAATCACTTACTCCTCCCACAGCGACTTAGGGCTGACTAACTTATTTTATTGCACGCCTTCTAGGGGGAG
TCTGCTGTCTTTCCAAATTATGCCAATTGTTTCCTAATCATTTGTATACCTATATATGTGTGTGTATATATACATATTTAATATATAACGTAATCCTGAAAATAAATGTATTTTATTGCAGCAAGGATTCTGTCATCAGTAGACACATAGGAAACGATTTTTCTTCTGTATCTCTGTACCTTCCTTA
CATGGTATAAAGGATGAAAGAATAGGTTAGAGATGAAAAACAGCACCCTTAAATACAGTCTGAACCTCATGATTTCTAAACCTTGGGAGCGCATGTGTGCCTTCACCCTTCTCTTTCTCTGGGCTTGCTCTTCTTAAG
```

* log => 상황 기록

** output info

contigs output by MEGAHIT

Brandon Seah edited this page on 14 Jan 2019 · 4 revisions

The FASTA file **final.contigs.fa** (or **OUTPUT_PREFIX.contigs.fa** if `--out-prefix` is specified) is the final result of the assembly.

For files in the folder **intermediate_contigs**:

- k***K*.contigs.fa** contains the contigs assembled from the de Bruijn graph of order-***K***, they can be converted to a SPAdes-like FASTG file for visualization
- k***K*.addi.fa** contains the contigs assembled after iteratively removing local low coverage *unitigs* in the de Bruijn graph of order-***K***
- k***K*.local.fa** contains the locally assembled contigs for k=***K***
- k***K*.final.contigs.fa** contains the stand-alone contigs for k=***K***; if local assembly is turned on, the file will be empty

> https://github.com/voutcn/megahit/wiki/contigs-output-by-MEGAHIT

>> github에서 wiki 파트에 이런 좋은 정보 많음