Prodigal

2022-02-04

20193852 문유빈

1) Goal : Prodigal로 CDS(coding sequence) 찾기

2) Prodigal

▷ Introduction:

==

Fast, reliable protein-coding gene prediction for prokaryotic genomes.

Contig 파일에서 어떤 부분이 CDS, ORF, RNA 등인지 찾아주는 프로그램

▷ What does Prodigal do?

==

(1) Predicts protein-coding genes
(2) Handles draft genomes and metagenomes
(3) Runs quickly
(4) Runs unsupervised
(5) Handles gaps, scaffolds, and partial genes
(6) Identifies translation initiation sites
(7) Outputs detailed summary statistics for each genome

** 자세한 설명은 prodigal github wiki에 서술돼 있음
https://github.com/hyattpd/prodigal/wiki/Introduction

▷ Parameter:

```
Usage:  prodigal [-a trans_file] [-c] [-d nuc_file] [-f output_type]
                 [-g tr_table] [-h] [-i input_file] [-m] [-n] [-o output_file]
                 [-p mode] [-q] [-s start_file] [-t training_file] [-v]

         -a:  Write protein translations to the selected file.
                > samples.faa 파일
         -c:  Closed ends.  Do not allow genes to run off edges.
         -d:  Write nucleotide sequences of genes to the selected file.
         -f:  Select output format (gbk, gff, or sco).  Default is gbk.
                > gff 파일
         -g:  Specify a translation table to use (default 11).
         -h:  Print help menu and exit.
```

-i: Specify FASTA/Genbank input file (default reads from stdin).

> contig.fa (assembly한 파일 넣기)

-m: Treat runs of N as masked sequence; don't build genes across them.

-n: Bypass Shine-Dalgarno trainer and force a full motif scan.

-o: Specify output file (default writes to stdout).

> samples.gff

-p: Select procedure (single or meta). Default is single.

> meta (우린 metagenome이니 meta로 돌리기)

-q: Run quietly (suppress normal stderr output).

-s: Write all potential genes (with scores) to the selected file.

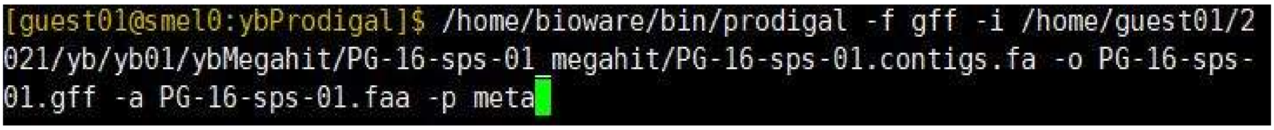
-t: Write a training file (if none exists); otherwise, read and use the specified training file.

-v: Print version number and exit.

3) 입력 코드

① ② ③ ④ ⑤

/home/bioware/bin/prodigal -f gff -i sample.contig.fa -o samples.gff -a sample.faa -p meta ⑥



```
[guest01@smel0:ybProdigal]$ /home/bioware/bin/prodigal -f gff -i /home/guest01/2
021/yb/yb01/ybMegahit/PG-16-sps-01_megahit/PG-16-sps-01.contigs.fa -o PG-16-sps-
01.gff -a PG-16-sps-01.faa -p meta
```

① /home/bioware/bin/prodigal => prodigal 프로그램 경로

② -f gff => output 형식을 gff format으로 설정

** gff = Generic Feature Format Version 3 output.

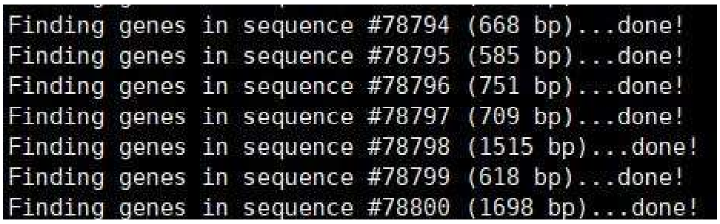
③ -i /home/guest01/2021/yb/yb01/ybMegahit/PG-16-sps-01_megahit/PG-16-sps-01.contigs.fa => input으로 넣어 줄 contig 파일의 경로

④ -o PG-16-sps-01.gff => output file 1 (gff 파일)

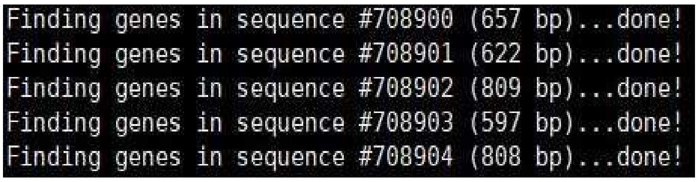
⑤ -a PG-16-sps-01.faa => output file 2 (faa 파일)

⑥ -p meta => metagenome 모드

4) 경과



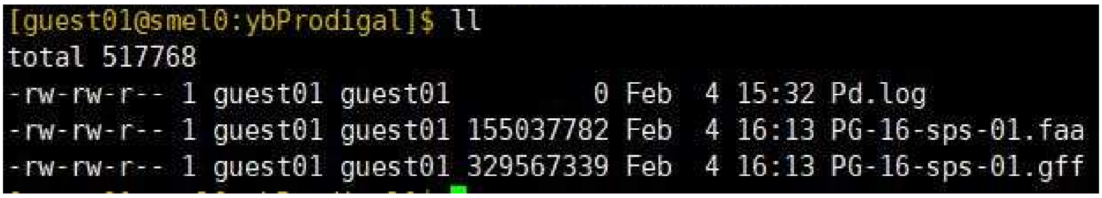
```
Finding genes in sequence #78794 (668 bp)...done!
Finding genes in sequence #78795 (585 bp)...done!
Finding genes in sequence #78796 (751 bp)...done!
Finding genes in sequence #78797 (709 bp)...done!
Finding genes in sequence #78798 (1515 bp)...done!
Finding genes in sequence #78799 (618 bp)...done!
Finding genes in sequence #78800 (1698 bp)...done!
```



```
Finding genes in sequence #708900 (657 bp)...done!
Finding genes in sequence #708901 (622 bp)...done!
Finding genes in sequence #708902 (809 bp)...done!
Finding genes in sequence #708903 (597 bp)...done!
Finding genes in sequence #708904 (808 bp)...done!
```

진행 중
.
.
.
.
완료

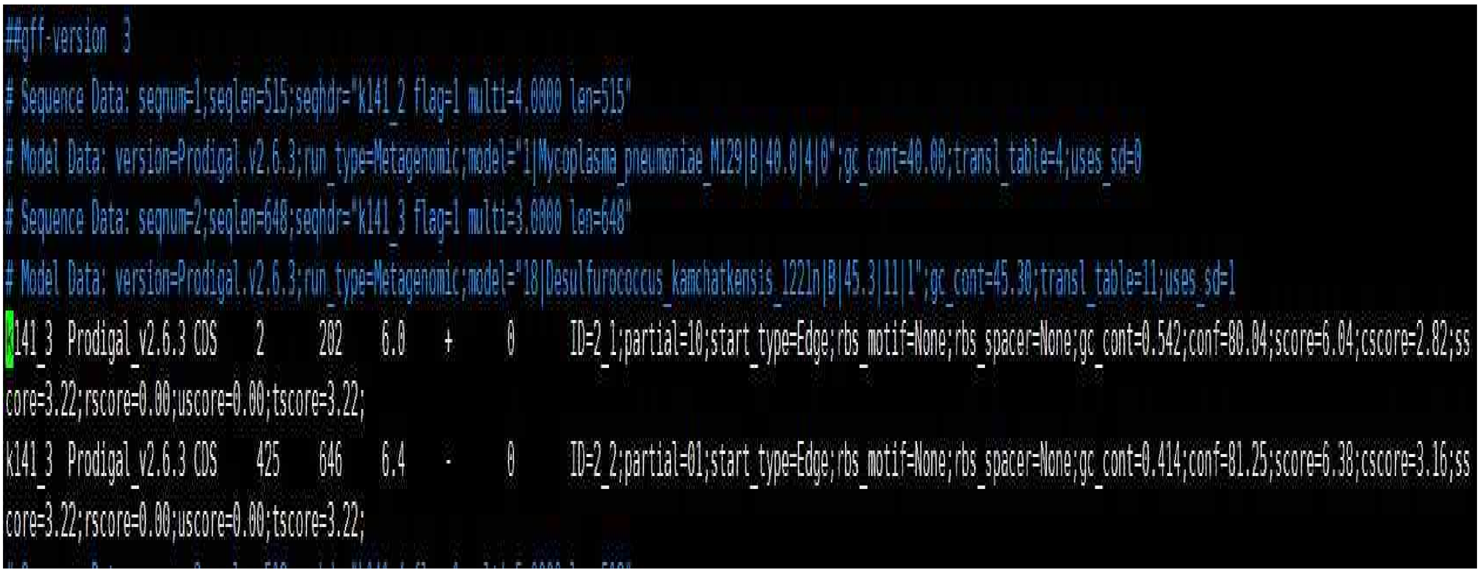
5) output



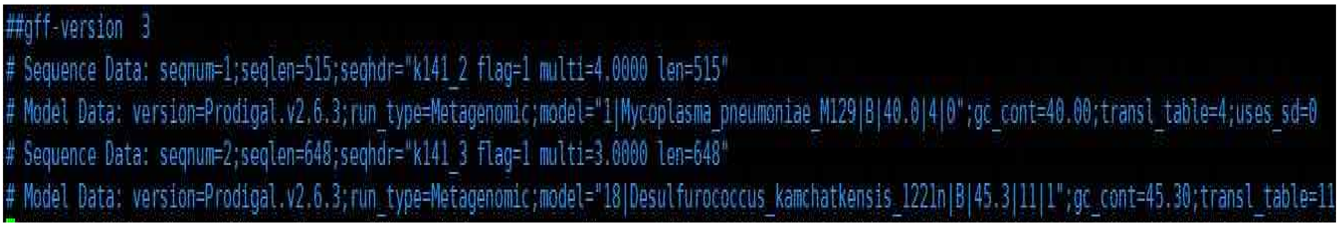
```
[guest01@smel0:ybProdigal]$ ll
total 517768
-rw-rw-r-- 1 guest01 guest01         0 Feb  4 15:32 Pd.log
-rw-rw-r-- 1 guest01 guest01 155037782 Feb  4 16:13 PG-16-sps-01.faa
-rw-rw-r-- 1 guest01 guest01 329567339 Feb  4 16:13 PG-16-sps-01.gff
```

** faa 파일 1, gff 파일 1 생성됨

5-1) PG-16-sps-01.gff



```
##gff-version  3
# Sequence Data: seqnum=1;seqlen=515;seqhdr="k141_2 flag=1 multi=4.0000 len=515"
# Model Data: version=Prodigal.v2.6.3;run_type=Metagenomic;model="1|Mycoplasma_pneumoniae_M129|B|40.0|4|0";gc_cont=40.00;transl_table=4;uses_sd=0
# Sequence Data: seqnum=2;seqlen=648;seqhdr="k141_3 flag=1 multi=3.0000 len=648"
# Model Data: version=Prodigal.v2.6.3;run_type=Metagenomic;model="18|Desulfurococcus_kamchatkensis_1221n|B|45.3|11|1";gc_cont=45.30;transl_table=11;uses_sd=1
k141_3  Prodigal_v2.6.3 CDS     2       202     6.0     +       0       ID=2_1;partial=10;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.542;conf=80.04;score=6.04;cscore=2.82;sscore=3.22;rscore=0.00;uscore=0.00;tscore=3.22;
k141_3  Prodigal_v2.6.3 CDS     425     646     6.4     -       0       ID=2_2;partial=01;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.414;conf=81.25;score=6.38;cscore=3.16;sscore=3.22;rscore=0.00;uscore=0.00;tscore=3.22;
```



```
##gff-version  3
# Sequence Data: seqnum=1;seqlen=515;seqhdr="k141_2 flag=1 multi=4.0000 len=515"
# Model Data: version=Prodigal.v2.6.3;run_type=Metagenomic;model="1|Mycoplasma_pneumoniae_M129|B|40.0|4|0";gc_cont=40.00;transl_table=4;uses_sd=0
# Sequence Data: seqnum=2;seqlen=648;seqhdr="k141_3 flag=1 multi=3.0000 len=648"
# Model Data: version=Prodigal.v2.6.3;run_type=Metagenomic;model="18|Desulfurococcus_kamchatkensis_1221n|B|45.3|11|1";gc_cont=45.30;transl_table=11
```

Sequence Data:

◎seqnum=1; => Ordinal ID for this sequence, beginning at 1.

◎seqlen=515; => Number of bases in the sequence

◎seqhdr="k141_2 ... ~~~" => Entire FASTA header line

Model Data:

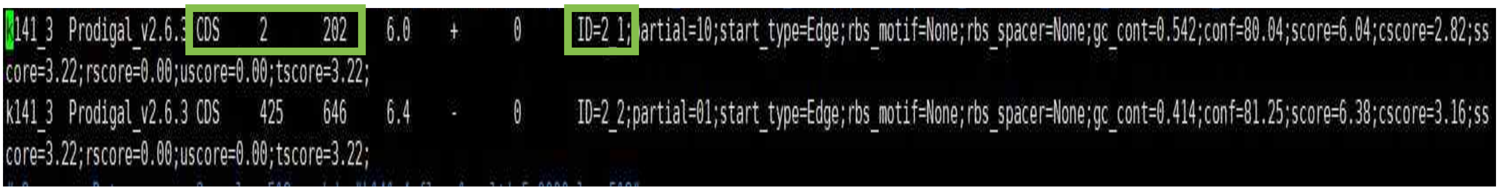
◎run_type=Metagenomic; => metagenome mode

◎model="1|Myco ... ~~~"; => Information about the preset training file used to analyze the sequence

◎gc_cont=40.00; => % GC content of the sequence

◎transl_table=4; => The genetic code used to analyze the sequence

◎used_sd=0 => Set to 1 if Prodigal used its default RBS finder, 0 if it scanned for other motifs.



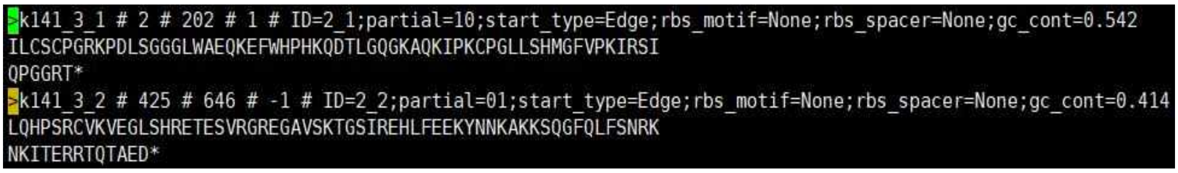
```
k141_3  Prodigal_v2.6.3 CDS     2       202     6.0     +       0       ID=2_1;partial=10;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.542;conf=80.04;score=6.04;cscore=2.82;ss
core=3.22;rscore=0.00;uscore=0.00;tscore=3.22;
k141_3  Prodigal_v2.6.3 CDS     425     646     6.4     -       0       ID=2_2;partial=01;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.414;conf=81.25;score=6.38;cscore=3.16;ss
core=3.22;rscore=0.00;uscore=0.00;tscore=3.22;
```

* 각각 무엇인지 prodigal wiki에 자세히 나와 있음
https://github.com/hyattpd/prodigal/wiki/understanding-the-prodigal-output#summary-statistics

** 우리가 알아야 하는 것

CDS 2 202 => Coding sequence가 2번 ~ 202번 (이 부분 추출해야 함)
ID=2_1 => ORF name

5-2) PG-16-sps-01.faa



```
>k141_3_1 # 2 # 202 # 1 # ID=2_1;partial=10;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.542
ILCSCPGRKPDLSGGGLWAEQKEFWHPHKQDTLGQGKAQKIPKCPGLLSHMGFVPKIRSI
QPGGRT*
>k141_3_2 # 425 # 646 # -1 # ID=2_2;partial=01;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.414
LQHPSRCVKVEGLSHRETESVRGREGAVSKTGSIREHLFEEKYNNKAKKSQGFQLFSNRK
NKITERRTQTAED*
```

① # 2 # 202 => 2번부터 202번 까지

② # ID=2_1; => gff 파일과 동일

③ ILCSCPGRKPDLSGGGLWAEQKEFWHPHKQDTLGQGKAQKIPKCPGLLSHMGFVPKIRSI QPGGRT* 해당 시퀀스