

Introduction to analyzing FASTA files version 2

Frank Aylward

Abstract

This is a short tutorial on how to get started analyzing FASTA files via the command line.

Code is intended for use on an Ubuntu 16.04 LTS OS, but it may work on other Unix or Unix-like systems.

Here we will mainly use standard Unix commands as well as the seqtk tool.

Information on seqtk, including information on installation, can be found here: <https://github.com/lh3/seqtk>

Citation: Frank Aylward Introduction to analyzing FASTA files. **protocols.io**

[dx.doi.org/10.17504/protocols.io.qg6dtze](https://doi.org/10.17504/protocols.io.qg6dtze)

Published: 28 May 2018

Protocol

Finding the data

Step 1.

The first thing we need to do is get some data. For purposes here we'll work with some FASTA files from the genome sequencing project of *Yersinia pestis*, the causative agent of the plague.

A good place to go for genome data is the National Center for Biotechnology Information, or NCBI. Among other things, they post FTP sites with genomic data for many different organisms. Here is the FTP site for the *Yersinia pestis* strain CO92:

ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1

You can also see some more general information at this URL:

<https://www.ncbi.nlm.nih.gov/genome/?term=yersinia+pestis>

If you copy and paste the first URL into your browser you should see various files with different

extensions like .faa.gz or .fna.gz. The .gz here is the gzip extension, which indicates that these files have been compressed using gzip. The other extensions generally give some information about the kind of sequences that are inside. For example, FNA generally stands for 'FASTA nucleic acid' while FAA generally stands for 'FASTA amino acid'.

Setting up a working folder/directory

Step 2.

Now before the download anything I usually like to set up a new folder that I can work in. This is because I want to be able to easily see what new files have been created or downloaded through this process, and if I'm working on my Desktop or Downloads folder it is easy to get confused with all of the other files that are there. So let's create a simple directory first and then navigate into it. The commands for this are:

```
mkdir fasta_parsing
```

```
cd fasta_parsing
```

The first command "mkdir" will make the directory, and the second command "cd" will allow us to navigate into that directory. If you ever want to see what files are present in the directory, you can use the "ls" command. There shouldn't be anything in the directory now, so your commands should lead to something like this:

```
faylward@Aylward: ~/fasta_parsing
faylward@Aylward:~$
faylward@Aylward:~$
faylward@Aylward:~$
faylward@Aylward:~$
faylward@Aylward:~$
faylward@Aylward:~$ mkdir fasta_parsing
faylward@Aylward:~$ cd fasta_parsing
faylward@Aylward:~/fasta_parsing$ ls
faylward@Aylward:~/fasta_parsing$
```

Not particularly exciting, but at least there are no error messages :)

Downloading the data with wget

Step 3.

Now we can get started with some data that we saw from in first step. For starters let's download the raw nucleic acid genome sequence, which is in the GCF_000009065.1_ASM906v1_genomic.fna.gz file. To download this directly from the command line we can use the 'wget' command, which is part of the basic Ubuntu command line and should already be installed. The command below should download the file:

wget -O

y_pestis_genome.fna.gz **ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1/GCF_000009065.1_ASM906v1_genomic.fna.gz**

In the command above, the -O flag lets us specify what name we want the downloaded file to have. After that we just have to provide the full URL for the file (this can usually be achieved by right-clicking on the file in the FTP site and then clicking 'copy link address'). The wget command will give us a progress report and log while the download is happening.

Since this file is compressed, we will want to uncompress it with the 'gunzip' command:

gunzip y_pestis_genome.fna.gz

And then we can check what we have in the directory with the 'ls' command. You should see something like this:

```
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ wget -O y_pestis_genome.fna.gz ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1/GCF_000009065.1_ASM906v1_genomic.fna.gz
--2018-05-26 16:05:48-- ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1/GCF_000009065.1_ASM906v1_genomic.fna.gz
=> 'y_pestis_genome.fna.gz'
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.10, 2607:f220:41e:250::7
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.10|:21... connected.
Logging in as anonymous ... Logged in!
=> SYST ... done.      => PWD ... done.
=> TYPE I ... done.    => CWD (1) /genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1 ... done.
=> SIZE GCF_000009065.1_ASM906v1_genomic.fna.gz ... 1438012
=> PASV ... done.      => RETR GCF_000009065.1_ASM906v1_genomic.fna.gz ... done.
Length: 1438012 (1.4M) (unauthoritative)

GCF_000009065.1_ASM906v1_genomic. 100%[=====>] 1.37M 2.32MB/s in 0.6s
2018-05-26 16:05:50 (2.32 MB/s) - 'y_pestis_genome.fna.gz' saved [1438012]

faylward@Aylward:~/fasta_parsing$ gunzip y_pestis_genome.fna.gz
faylward@Aylward:~/fasta_parsing$ ls
y_pestis_genome.fna
faylward@Aylward:~/fasta_parsing$
```

If you see something like this, then success! We have our very own FASTA file that we can now begin to analyze.

The format and content of FASTA genome files

Step 4.

Typically whenever you have a new file that you want to look at, the first few commands you want to

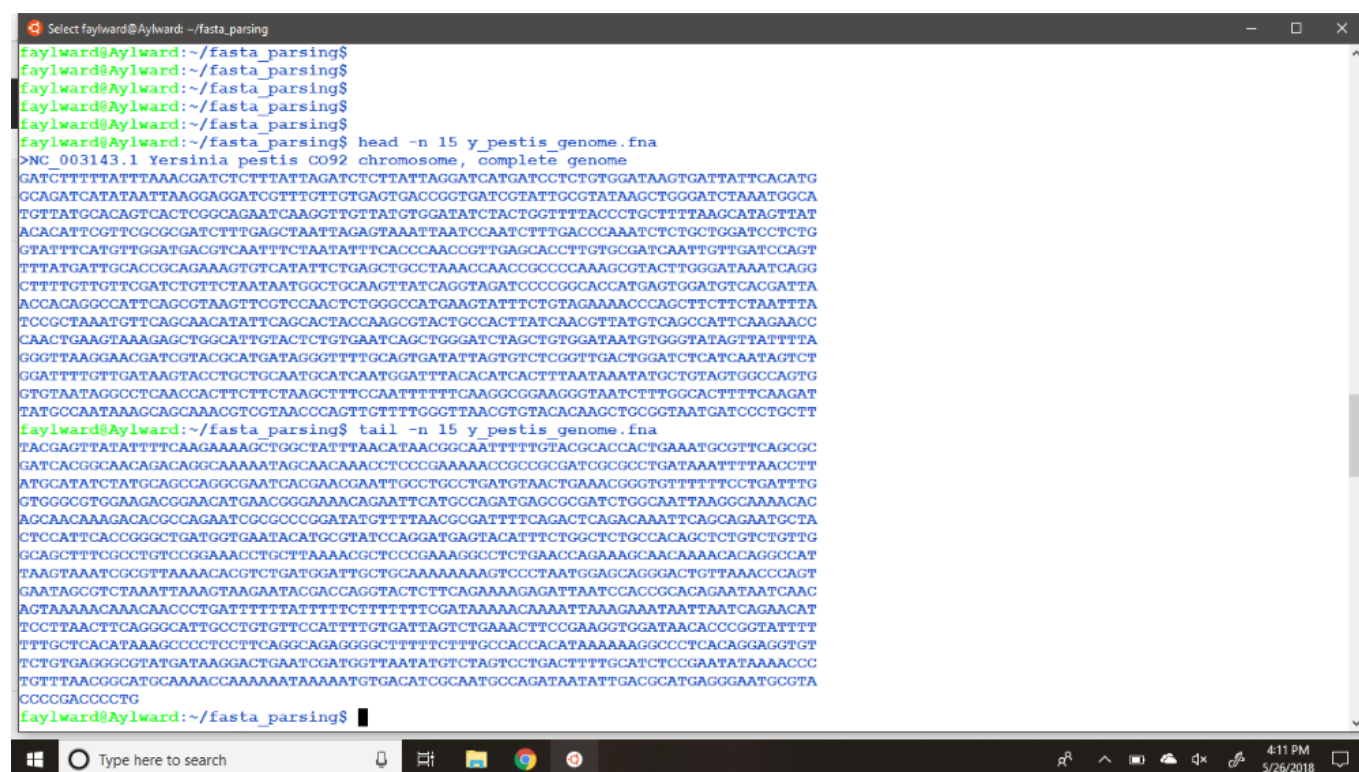
use are 'head', and 'tail' commands to look at the format of the beginning and end of the file. 'head' and 'tail' return the first and last 10 lines of a file, though we can specify the number of lines we want to see with the '-n' flag. Try something like this:

```
head -n 20 y_pestis_genome.fna
```

and

```
tail -n 20 y_pestis_genome.fna
```

and you should see something like this:



```
Select faylward@Aylward: ~/fasta_parsing
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ head -n 15 y_pestis_genome.fna
>NC_003143.1 Yersinia pestis C092 chromosome, complete genome
GATCTTTTATTTAAACGATCTCTTTATTAGATCTCTTATTAGGATCATGATCCTCTGCGATAAGTGATTATTCACATG
GCAGATCATATAAATTAAGGAGGATCGTTTGTGTGAGTGACCGGTGATCGTATTGCGGTATAAGCTGGGATCTAAATGGCA
TGTATGACAGCTCACTCGGCAGAAATCAAGGTTGTATGTGGATATCTACTGTTTACCGTCTTTTAAAGCATAGTTAT
ACACATTCGTTGCGCGATCTTTGAGCTAATTAGAGTAAATTAATCCAATCTTTGACCCAAATCTCTGCTGGATCCTCTG
GTATTTTCATGTTGGATGAGTCAATTTCTAATATTTACCCCAACCGTTGAGCACCTTTGTGCGATCAATGTTGATCCAGT
TTTATGATTGACCGCAGAAAGTGTCATATTCGAGCTGCCATAAACCAACCGCCCAAGCGTACTTGGGATAAATCAGG
CTTTTGTGTCGATCTGTTCTAATAATGGCTGCAAGTATCAGGTAGATCCCGGGCAGCATGAGTGGATGTCACGATTA
ACACAGGCCATTGAGCGTAAGTTCGTCCTCACTCGGCCATGAAGTATTTCTGTAGAAAACCGAGCTTCTTCTAATTTA
TCCGCTAAATGTTGAGCAACATATTCAGCACTACCAAGCGTACTGCCACTTATCAACGTTATGTCAGCCATTCAAGAACC
CAACTGAAGTAAGAGCTGGCATTGACTCTGGAATCAGCTGGGATCTAGCTGTGGATAATGTGGGTATAGTTATTTTA
GGGTTAAGGAACGATCGTACGATGATAGGGTTTTCAGTGATATTAGTGTCTCGGTTGACTGGATCTCATCAATAGTCT
GGATTTTGTGATAAGTACCTGCTGCAATGCATCAATGGATTTACACATCACTTTAATAAATATGCTGTAGTGGCCAGTG
GTGTAATAGGCTCAACCACTTCTTCAAGCTTTTCAATTTTTCAGGCGGAAGGGTAATCTTTGGCACTTTTCAAGAT
TATGCCAATAAAGCAGCAACCTCGTAACCCAGTCTTTTGGGTTAACGTGTACACAGCTCGCGTAATGATCCCTGCTT
faylward@Aylward:~/fasta_parsing$ tail -n 15 y_pestis_genome.fna
TACGATCTTATATTTTCAAGAAAGTGGCTATTTAACAATAACGGCAATTTTGTACGCACCACTGAAATGCGTTTACGGC
GATCACGGCAACACAGCGCAAAATAGCAACAAACCTCCCGAAAAACCGCGGATCGCGCTGATAAATTTTAACTT
ATGCATATCTATGACGCGAGCGGAATCAGGAACGAATTCCTGCTGATGTAAGTGAAGCGGTGTTTTTCTGATTG
TGCGCGGTGGAGCGGAACATGAACGGGAAAAAGCAATTCATGCCAGATGACGCGGATCTGGCAATTAAGGCAAAACAC
AGCAACAAAGACAGCGCAGAAATCGGGCCCGGATATGTTTAAACGGATTTTACAGACTCAGACAAATTCAGCAGAATGCTA
CTCCATTACCGGGCTGATGGTGAATACATGCGTATCCAGGATGAGTACATTTCTGGCTCTGCCACAGCTCTGTCTGTTG
CGAGCTTTGCGCTGTCGGGAAACCTGCTTAAACGCTCCCGAAAGGCTCTGAACAGAAAGCAACAAACACAGGCCAT
TAAGTAATTCGGCTTAAACACGCTGATGGATTGCTGCAAAAAAAGTCCCTAATGGAGCAGGACTGTTTAAACCCAGT
GAATAGCTCTAAATTAAGTAAAGTACGACAGGTACTCTTCAGAAAAGAGATTAATCCACCCACAGAAATATCAAC
AGTAAAAACAAACACCGCTGATTTTATTTTCTTTTTCGATAAAAGCAAAATTAAGAAATTAATTAATCAGAAGAT
TCCTTAACCTTCAAGGCTATGCTGCTGTTCCATTTTCTGATTAGTCTGAAACTTCCGAAAGTGGATAACACCCGGTATTT
TTTGCTCACATAAAGCCCTCTTCAGGCAGAGGGGCTTTTCTTTGCCACCACATAAAAAAGGCCCTCACAGGAGGTG
TCTGTGAGGCGTATGATAAGGACTGAATCGATGGTTAATATGTCTAGTCTGACTTTTGCATCTCCGAATATAAAACCC
TGTTTAAACGGCATGCAAAACCAAAAAATAAAATGTGACATCGCAATGCCAGATAAATTAGCGCATGAGGGAATCGGTA
CCCCGACCCCTG
faylward@Aylward:~/fasta_parsing$
```

So you essentially see that the file is full of ATGC letters, and that the very beginning has a header line with some descriptive information that starts with a '>'. This is the basic format of FASTA files. Generally what comes right after the '>' in the header line should be a non-redundant identifier for the sequence that follows it. Also, there can be multiple sequences in a single file, all of which have a header line that precedes the sequence.

Now, you may ask, why bother looking at a file in the command line if we can just open it up in our favorite text editor, like gedit or notepad++? Well, opening up the file in a text editor is perfectly fine, and if you're just starting you may wish to do this just to browse, but using the command line has two main advantages: 1) You can quickly check the format and contents of a file, so you don't have to

toggle between different programs, which is especially handy if you're working with many different files, and 2) FASTA files can get huge (giga- or terabyte size), and trying to open those files in your text editor will either take a very long time or crash your system.

If you want to have more open-ended browsing capabilities of a file, you can also try the "less" command, which allows you to browse through a file.

Analyzing genome files with grep

Step 5.

Now that we have a general idea of what the FASTA file looks like, we can get some basic statistics about how many distinct sequences are inside.

Remember that above I indicated that a single FASTA file can have multiple distinct sequences inside (i.e., multiple chromosomes, plasmids, or whole genomes could be present in a single file, with each unique nucleic acid sequence preceded by its own '>' header line).

So how many distinct sequences are in our *Y. pestis* genome? We can use the 'grep' command for this:

```
grep '^>' y_pestis_genome.fna
```

and

```
grep -c '^>' y_pestis_genome.fna
```

Grep searches through a file and returns all of the lines that have a particular character or pattern. Here we searched for '^>', which tells grep that we want every line that starts with a '>' character (the '^' is code for 'begins with'). Since each FASTA header starts with a '>' character, this should return all of the headers for each unique sequence in the file.

In the second command we did the same thing, but with the '-c' flag, which tells grep we don't actually want to see each line that has the query character in it, we just want to know how many total lines that fit our search.

The results should look like this:

```
faylward@Aylward:~/fasta_parsing$ grep "^>" y_pestis_genome.fna
>NC_003143.1 Yersinia pestis CO92 chromosome, complete genome
>NC_003131.1 Yersinia pestis CO92 plasmid pCD1, complete sequence
>NC_003134.1 Yersinia pestis CO92 plasmid pMT1, complete sequence
>NC_003132.1 Yersinia pestis CO92 plasmid pPCP1, complete sequence
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ grep -c "^>" y_pestis_genome.fna
4
faylward@Aylward:~/fasta_parsing$
```

The descriptions in the headers tell us quite a bit about the sequences that are in our file. Each header starts with a unique identifier (NC_#####), followed by description. The first sequence therefore encodes the *Yersinia pestis* chromosome, and the next three sequences encode for plasmids.

Using the '-c' command, we can quickly see that there are 4 total sequences present.

Downloading FASTA files of genes

Step 6.

This is all good and well for looking at a raw genome file, but what about the other files we saw in step?

Let's continue with analyzing the genes encoded in the *Y. pestis* genome. These will be found in the 'GCF_000009065.1_ASM906v1_cds_from_genomic.fna.gz' file.

Note that the file name includes the abbreviation "CDS", which means "coding sequence" and implies that these are protein-coding genes. So other genes, such as rRNAs or tRNAs, are not going to be found here.

Let's download and uncompress the data the same as with the genome.

wget -O y_pestis_genes.fna.gz

ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1/GCF_000009065.1_ASM906v1_cds_from_genomic.fna.gz

gunzip y_pestis_genes.fna

ls

And you should see both the genome and gene files:

```

faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ wget -O y_pestis_genes.fna.gz ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1/GCF_000009065.1_ASM906v1_ods_from_genomic.fna.gz
--2018-05-26 16:36:41-- ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1/GCF_000009065.1_ASM906v1_ods_from_genomic.fna.gz
      => 'y_pestis_genes.fna.gz'
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.13, 2607:f220:41e:250::13
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.13|:21... connected.
Logging in as anonymous ... Logged in!
=> SYST ... done.      => PWD ... done.
=> TYPE I ... done.    => CWD (1) /genomes/all/GCF/000/009/065/GCF_000009065.1_ASM906v1 ... done.
=> SIZE GCF_000009065.1_ASM906v1_ods_from_genomic.fna.gz ... 1414093
=> PASV ... done.      => RETR GCF_000009065.1_ASM906v1_ods_from_genomic.fna.gz ... done.
Length: 1414093 (1.3M) (unauthoritative)

GCF_000009065.1_ASM906v1_ods_from 100%[>] 1.35M 2.62MB/s in 0.5s

2018-05-26 16:36:42 (2.62 MB/s) - 'y_pestis_genes.fna.gz' saved [1414093]

faylward@Aylward:~/fasta_parsing$ gunzip y_pestis_genes.fna.gz
faylward@Aylward:~/fasta_parsing$ ls
y_pestis_genes.fna y_pestis_genome.fna
faylward@Aylward:~/fasta_parsing$

```

Analyzing FASTA files of genes

Step 7.

Now we can use the same grep commands with the 'genes' file as we used with the 'genome' file, to get an idea of the format and the number of genes encoded in the genome.

head y_pestis_genes.fna

```

faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ head y_pestis_genes.fna
>lcl|NC_003143.1.ods.YP_002345097.1.1 [locus_tag=YPO0001] [db_xref=InterPro:IPR001094,InterPro:IPR008254,UniProtKB/TrEMBL:Q7CLEF0] [protein=flavodoxin] [protein_id=YP_002345097.1] [location=complement(271..711)] [gbkey=CDS]
ATGGCTGACATAACGTTGATAAGTGGCAGTACGCTTGGTAGTGCTGAATATGTTGCTGAACATTTAGCGGATAAAATTAGA
AGAAGCTGGGTTTTCTACAGAAATACTTCATGGCCAGAGTTGGACGAACCTACGCTGAATGGCCTGTGGTTAATCGTGA
TCCCACTCATGGTCCCGGGGATCTACCTGATAACTTGCAGCCATTATTAGAACAGATCGAACAACAAAAGCCTGATTTA
TCCCAAGTACGCTTTGGGGCGGTTGGTTTAGGCAGCTCAGAAATATGACACTTTCTGCGGTGCAATCAAAAAGTGGATCA
ACAAATGATCGGCACAAGGTGCTCAACGGTTGGGTGAAATATTAGAAATTGACGTCATCCAACATGAAATACCAGAGGATC
CAGCAGAGATTTGGGTCAAAGATTGGATTAATTTACTCTAA
>lcl|NC_003143.1.ods.YP_002345098.1.2 [gene=asnC] [locus_tag=YPO0002] [db_xref=InterPro:IPR000485,InterPro:IPR002197,InterPro:IPR0110
08,InterPro:IPR011991,UniProtKB/TrEMBL:Q7CLE9] [protein=DNA-binding transcriptional regulator AsnC] [protein_id=YP_002345098.1] [locat
ion=complement(804..1265)] [gbkey=CDS]
ATGAGTGAAATTTATCAGATCGATAATCTCGATCGCAGCATCCTGAAAGCATTAAATGGAAATGCACGCACACCCATATGC
TGAATTAGCCAAAACCTCGCTGTTAGCCCCGGAACATTTCATGTAAGAGTAGAAAAGATGCGGCAAGCAGGGATCATTA
faylward@Aylward:~/fasta_parsing$

```

Note that in the first 10 lines we can already see multiple header lines, since of course genes are much shorter than whole genomes.

Also note that the headers for genes have a lot more information. In addition to unique identifiers, they also have information about the function of the gene, the coordinates of the gene in the chromosome/plasmid, a unique identifier for the protein that is encoded, and other information that NCBI keeps in its databases.

We can continue to take a look at the file with grep:

grep '^>' y_pestis_genes.fna | head

grep -c '^>' y_pestis_genes.fna

```
aylward@Aylward:~/fasta_parsing$
aylward@Aylward:~/fasta_parsing$ grep '^>' y_pestis_genes.fna | head
>|l|NC_003143.1_cds_YP_002345097.1_1 [locus_tag=YPO0001] [db_xref=InterPro:IPR001094,InterPro:IPR008254,UniProtKB/TrEMBL:Q7CLF0] [pr
otein=flavodoxin] [protein_id=YP_002345097.1] [location=complement(271..711)] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345098.1_2 [gene=asnC] [locus_tag=YPO0002] [db_xref=InterPro:IPR000485,InterPro:IPR002197,InterPro:IPR0110
08,InterPro:IPR011991,UniProtKB/TrEMBL:Q7CLE9] [protein=DNA-binding transcriptional regulator AsnC] [protein_id=YP_002345098.1] [loca
tion=complement(804..1265)] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345099.1_3 [gene=asnA] [locus_tag=YPO0003] [db_xref=InterPro:IPR004618,InterPro:IPR006195,UniProtKB/TrEMBL
:Q0WKT9] [protein=asparagine synthetase AsnA] [protein_id=YP_002345099.1] [location=1435..2427] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345100.1_4 [gene=yieM] [locus_tag=YPO0004] [db_xref=InterPro:IPR002035,UniProtKB/TrEMBL:Q0WKT8] [protein=h
ypothetical protein] [protein_id=YP_002345100.1] [location=complement(2526..3992)] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345101.1_5 [locus_tag=YPO0005] [db_xref=InterPro:IPR001270,InterPro:IPR003593,InterPro:IPR011704,UniProtKB
/TrEMBL:Q0WKT7] [protein=regulatory ATPase Rava] [protein_id=YP_002345101.1] [location=complement(3996..5549)] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345102.1_6 [gene=trkD] [locus_tag=YPO0006] [db_xref=InterPro:IPR003855,UniProtKB/TrEMBL:Q0WKT6] [protein=p
otassium transport protein Kup] [protein_id=YP_002345102.1] [location=5823..7691] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345103.1_7 [locus_tag=YPO0007] [db_xref=InterPro:IPR007721,UniProtKB/TrEMBL:Q7CLE5] [protein=D-ribose pyra
nase] [protein_id=YP_002345103.1] [location=7896..8315] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345104.1_8 [gene=rbsK] [locus_tag=YPO0008] [db_xref=InterPro:IPR002139,InterPro:IPR002173,InterPro:IPR0116
11,InterPro:IPR011877,UniProtKB/TrEMBL:Q7CLE4] [protein=ribokinase] [protein_id=YP_002345104.1] [location=8366..9292] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345105.1_9 [locus_tag=YPO0009] [db_xref=InterPro:IPR001411,InterPro:IPR007114,InterPro:IPR011701,UniProtKB
/TrEMBL:Q7CLE3] [protein=membrane transport protein] [protein_id=YP_002345105.1] [location=complement(9512..10936)] [gbkey=CDS]
>|l|NC_003143.1_cds_YP_002345106.1_10 [locus_tag=YPO0010] [db_xref=InterPro:IPR000524,InterPro:IPR011711,UniProtKB/TrEMBL:Q7CLE2] [p
rotein=GntR family transcriptional regulator] [protein_id=YP_002345106.1] [location=complement(11016..11705)] [gbkey=CDS]
aylward@Aylward:~/fasta_parsing$
aylward@Aylward:~/fasta_parsing$ grep -c '^>' y_pestis_genes.fna
3979
aylward@Aylward:~/fasta_parsing$
```

Note that above I used a 'pipe' with the first grep command. A pipe takes the output of the first command and feeds it into the command that comes after the '|'. This is handy here since there are many genes in the *Y. pestis* genome, and all of the text in the command line can be overwhelming. This way we can see the first 10 to get a general idea of the format (still a bit difficult to browse, since the headers are so long, but they give us a general idea).

Lastly, with the "-c" flag with grep we can see that there are a total of 3,979 protein-coding genes in this genome, including those encoded in the chromosome and plasmids.

More complex grep commands with pipes

Step 8.

We can use pipes and combine Unix commands to get more specific information. Let's say we want to know how many genes are on the main chromosome only. Well we know from above that the main chromosome has the unique identifier NC_003143.1, so we can combine grep commands to count how many headers have only that

grep "^>" y_pestis_genes.fna | grep -c "NC_003143.1"

The first grep command give us all of the header lines, and the second grep command parses through these and returns only the header lines with the chromosome ID in them.

Here I got 3798. In the last step we found that there are 3979 genes total, so >95% of all of the genes are encoded on the main chromosome. This is pretty typical, since plasmids are usually pretty small compared to chromosomes.

Note that we probably could have arrived at this answer using only one `grep` command, since we know each gene name has the chromosome ID in the beginning.

```
grep -c ">|NC_003143.1" y_pestis_genes.fna
```

But with the first command we have a bit more versatility. For example, we can also ask how many genes have the term "DNA polymerase" in the description:

```
grep "^>" y_pestis_genes.fna | grep -c "DNA polymerase"
```

(I got 12). Or may be we want an idea of genes involved in virulence:

```
grep "^>" y_pestis_genes.fna | grep -c "virulence"
```

(I got 7). For in depth annotations we will want to do more than simple keyword searches, but these are still easy ways to begin exploring the data and looking at what kinds of genes are present.

These examples also give you an idea of how powerful simple Unix commands can be. Since several commands can be easily combined with pipes, many different questions can be answered with a little creativity.

Using seqtk

Step 9.

Sometimes we want a bit more information than just the format of a FASTA file and the total number of sequences that are inside. For more information we can use the "seqtk" tool.

To install, type:

```
sudo apt install seqtk
```

You should be prompted to enter in your password, and then you will see some information about how the installation proceeds. For more information about this tool you can look at the github page: <https://github.com/lh3/seqtk>

To see what utilities are available in the seqtk tool, just type:

seqtk

```
faylward@Aylward:~/fasta_parsing$ seqtk

Usage:  seqtk <command> <arguments>
Version: 1.0-r31

Command: seq      common transformation of FASTA/Q
         comp     get the nucleotide composition of FASTA/Q
         sample   subsample sequences
         subseq   extract subsequences from FASTA/Q
         trimfq   trim FASTQ using the Phred algorithm

         hety     regional heterozygosity
         mutfa    point mutate FASTA at specified positions
         mergefa  merge two FASTA/Q files
         randbase choose a random base from hets
         cutN     cut sequence at long N
         listhet  extract the position of each het

faylward@Aylward:~/fasta_parsing$
```

You can see that there are many things we can use seqtk for. Here we will only be using the "comp" utility, which gives us some general statistics of sequences in a FASTA file.

To see what this does, try the following command:

seqtk comp y_pestis_genome.fna

```
faylward@Aylward:~/fasta_parsing$ seqtk comp y_pestis_genome.fna
NC_003143.1 4653728 1219520 1102670 1114185 1217353 0 0 0 524242 0 0 0
NC_003131.1 70305 19135 15863 15659 19648 0 0 0 6652 0 0 0
NC_003134.1 96210 23255 24651 23673 24631 0 0 0 13202 0 0 0
NC_003132.1 9612 2792 2253 2098 2469 0 0 0 966 0 0 0
faylward@Aylward:~/fasta_parsing$
```

Each row here is a unique sequence in the FASTA file, which for the Y. pestis genome would correspond to the chromosome and three plasmids. The columns provide the name, length, #A, #C, #G, #T, #2, #3, #4, #CpG, #tv, #ts, #CpG-ts.

There is a lot of information here about the composition of the sequences, but I'm usually only interested in the sequence length, ACGT composition, and possibly the #4 column, which specifies how many N characters were present in the sequence (if a nucleotide is ambiguous, meaning it could not be determined for some reason, it will sometimes be given a letter other than ATGC: see the one-letter codes here: <https://www.bioinformatics.org/sms/iupac.html>).

So from the above we can see that the first sequence, which is the

chromosome, is much longer than the following three sequences, which are all plasmids (>4 million base-pairs compared to 70K, 96K, and 10K). Plasmids are typically much shorter than chromosomes, so this is not surprising.

Analyzing genes with seqtk

Step 10.

Now let's use 'seqtk comp' on the genes file, once again using a pipe so we don't flood our command line with text:

seqtk comp y_pestis_genes.fna | head

```
faylward@Aylward:~/fasta_parsing$ seqtk comp y_pestis_genes.fna | head
lcl|NC_003143.1_cds_YP_002345097.1_1 441 134 82 107 118 0 0 0 28 0 0 0
lcl|NC_003143.1_cds_YP_002345098.1_2 462 153 102 90 117 0 0 0 32 0 0 0
lcl|NC_003143.1_cds_YP_002345099.1_3 993 267 190 262 274 0 0 0 76 0 0 0
lcl|NC_003143.1_cds_YP_002345100.1_4 1467 401 346 354 366 0 0 0 162 0 0 0
lcl|NC_003143.1_cds_YP_002345101.1_5 1554 456 382 343 373 0 0 0 122 0 0 0
lcl|NC_003143.1_cds_YP_002345102.1_6 1869 401 386 470 612 0 0 0 184 0 0 0
lcl|NC_003143.1_cds_YP_002345103.1_7 420 117 99 103 101 0 0 0 58 0 0 0
lcl|NC_003143.1_cds_YP_002345104.1_8 927 232 201 263 231 0 0 0 130 0 0 0
lcl|NC_003143.1_cds_YP_002345105.1_9 1425 309 319 343 454 0 0 0 136 0 0 0
lcl|NC_003143.1_cds_YP_002345106.1_10 690 203 144 160 183 0 0 0 58 0 0 0
faylward@Aylward:~/fasta_parsing$
```

Now let's say we don't want to know what the longest gene in the genome is. For this we can pipe the output of 'seqtk comp' into the 'sort' command:

seqtk comp y_pestis_genes.fna | sort -k 2,2 -rn | head

```
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ seqtk comp y_pestis_genes.fna | sort -k 2,2 -rn | head
lcl|NC_003143.1_cds_YP_002346049.1_921 11118 2645 2589 3295 2589 0 0 0 1504 0 0 0
lcl|NC_003143.1_cds_YP_002345671.1_558 9888 2359 2677 2982 1870 0 0 0 1648 0 0 0
lcl|NC_003143.1_cds_YP_002346901.1_1763 9492 1858 2930 2840 1864 0 0 0 2072 0 0 0
lcl|NC_003143.1_cds_YP_002345994.1_872 9333 3054 1479 1995 2805 0 0 0 512 0 0 0
lcl|NC_003143.1_cds_YP_002348815.1_3637 9042 2286 2144 2485 2127 0 0 0 1222 0 0 0
lcl|NC_003143.1_cds_YP_002347454.1_2298 7608 1845 2032 2330 1401 0 0 0 1230 0 0 0
lcl|NC_003143.1_cds_YP_002345834.1_719 6606 1541 1539 1769 1757 0 0 0 742 0 0 0
lcl|NC_003143.1_cds_YP_002346902.1_1764 6108 1199 1933 1741 1235 0 0 0 1236 0 0 0
lcl|NC_003143.1_cds_YP_002347532.1_2377 6015 1579 1507 1495 1434 0 0 0 704 0 0 0
lcl|NC_003143.1_cds_YP_002345832.1_717 5820 1493 1376 1491 1460 0 0 0 630 0 0 0
faylward@Aylward:~/fasta_parsing$
```

Just to unpack the syntax of 'sort' a bit, the '-r' flag indicates we wish to perform a reverse sort (descending order), the '-n' flag indicates we wish to perform a numeric sort (as opposed to alphabetical, which wouldn't make sense here), and the '-k' flag indicates the columns to use for sorting, which in this case is only the 2nd column. The default column delimiter for 'sort' is a tab, which is already the column delimiter used by 'seqtk comp' in its output, so we don't need to change anything there. If columns were separated with something else, we would want to use the '-d' flag (as with all Unix commands, check all of the features of sort by typing 'man sort').

So in the output above we have found that lcl|NC_003143.1_cds_YP_002346049.1_921 is the longest gene, with a length of 11,118, or a little more than 11 kilobases. A good rule-of-thumb for bacteria is that the average gene length of a genome is 1 kilobase, so this one is pretty large!

Now what if we want to find what this gene encodes for? Since we know that the FASTA header in the genes file has this information, we can use grep to retrieve this information:

grep 'lcl|NC_003143.1_cds_YP_002346049.1_921' y_pestis_genes.fna

```
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ seqtk comp y_pestis_genes.fna | sort -k 2,2 -rn | head
lcl|NC_003143.1_cds_YP_002346049.1_921 11118 2645 2589 3295 2589 0 0 0 1504 0 0 0
lcl|NC_003143.1_cds_YP_002345671.1_558 9888 2359 2677 2982 1870 0 0 0 1648 0 0 0
lcl|NC_003143.1_cds_YP_002346901.1_1763 9492 1858 2930 2840 1864 0 0 0 2072 0 0 0
lcl|NC_003143.1_cds_YP_002345994.1_872 9333 3054 1479 1995 2805 0 0 0 512 0 0 0
lcl|NC_003143.1_cds_YP_002348815.1_3637 9042 2286 2144 2485 2127 0 0 0 1222 0 0 0
lcl|NC_003143.1_cds_YP_002347454.1_2298 7608 1845 2032 2330 1401 0 0 0 1230 0 0 0
lcl|NC_003143.1_cds_YP_002345834.1_719 6606 1541 1539 1769 1757 0 0 0 742 0 0 0
lcl|NC_003143.1_cds_YP_002346902.1_1764 6108 1199 1933 1741 1235 0 0 0 1236 0 0 0
lcl|NC_003143.1_cds_YP_002347532.1_2377 6015 1579 1507 1495 1434 0 0 0 704 0 0 0
lcl|NC_003143.1_cds_YP_002345832.1_717 5820 1493 1376 1491 1460 0 0 0 630 0 0 0
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$
faylward@Aylward:~/fasta_parsing$ grep "lcl|NC_003143.1_cds_YP_002346049.1_921" y_pestis_genes.fna
>lcl|NC_003143.1_cds_YP_002346049.1_921 [gene=yapH] [locus_tag=YPO1004] [db_xref=InterPro:IPR004899,InterPro:IPR005546,InterPro:IPR006315,InterPro:IPR011049,InterPro:IPR011050,InterPro:IPR013425,UniProtKB/TrEMBL:Q0WI39] [protein=autotransporter protein] [protein_id=YP_002346049.1] [location=1119829..1130946] [gbkey=CDS]
faylward@Aylward:~/fasta_parsing$
```

So it appears this very long gene is *yapH*, which encodes for an autotransporter protein. Not super descriptive, but sometimes the functions of genes and their encoded proteins is unclear.

Now try doing the same, but with the shortest gene in the *Y. pestis* genome. What is that gene, and what is its predicted function?

More complex genome arithmetic

Step 11.

For simple math in the command line I like to use a tool called "datamash". If this tool is not installed you can install it with:

sudo apt install datamash

This tool is nice because it will let use do simple calculations like sums and means. For example, let's say we want to know the mean gene length of all genes in the *Y. pestis* genome. We can do this with the following command:

seqtk comp y_pestis_genes.fna | datamash mean 2

The syntax for datamash is "datamash <operation to perform> <column to use>"

So here we want to calculate the mean of all values in the second column, which we know from our

work above are the gene lengths.

Or if we only want to look at genes on the main chromosome:

seqtk comp y_pestis_genes.fna | grep "NC_003143.1" | datamash mean 2

```
faylward@Aylward:~/fasta_parsing$  
faylward@Aylward:~/fasta_parsing$ seqtk comp y_pestis_genes.fna | datamash mean 2  
968.18421713998  
faylward@Aylward:~/fasta_parsing$ seqtk comp y_pestis_genes.fna | grep "NC_003143.1" | datamash mean 2  
979.10979462875  
faylward@Aylward:~/fasta_parsing$ █
```

Rather interesting. The average gene length is around 1 kilobase in both cases, and genes on the main chromosome are slightly longer on average than all genes combined.

Now what is the total length of all protein coding genes on the main chromosome? We can answer this question with the following command:

seqtk comp y_pestis_genes.fna | grep "NC_003143.1" | datamash sum 2

I got a value of 3718659 for this. From using seqtk in step 9 above we know that the main chromosome has a length of 4653728, so if we calculate the ratio of base-pairs in all protein-coding sequences divided by the total number of base-pairs we arrive at 79.9%. This could be called the "coding density" of protein-coding genes. Note that we are not considering rRNA, tRNA, or other genes here, so the total coding density will be higher than this. Still, 80% is pretty high, indicating that the main chromosome is densely packed with genes.