Command line exercises with yeast

Ken Youens-Clark

Abstract

Yeast is a well-characterized genome due to its small size and historical significance in genetics. The website http://yeastgenome.org/ is a dedicated resource for yeast genomics.

Citation: Ken Youens-Clark Command line exercises with yeast. **protocols.io**
dx.doi.org/10.17504/protocols.io.fnebmbe
Published: 23 Aug 2016

Protocol

Step 1.

Cerevisiae chromosomes

Create a directory for 'yeast':

```
$ mkdir /work/yeast
$ cd !$
```

Step 2.

Go to http://downloads.yeastgenome.org/sequence/S288C_reference/chromosomes/fasta/ and download the ".fsa" files. You can right-click on the links to copy the link location and then "wget" the file.

Can you think of a way to easily script this, or will you just click on all 17 chromosomes?

Step 3.

Make a single whole genome file called "cerevisiae_genome.fasta"

Step 4.

Count the chromosomes in the whole genome file using commands from the lecture. (HINT: Each of the original FASTA files contains a single chromosome).

Step 5.

Look up the command 'wc' and find out what it does. Get size of total genome. (HINT: The size of the genome can be determined by counting the number of characters not on the same line as a fasta header).

Step 6.

Cerevisiae genes

Get the list of cerevisiae chromosome features:
http://downloads.yeastgenome.org/curation/chromosomal_feature/SGD_features.tab

```
Columns within SGD_features.tab:

1.   Primary Standfor Gene Database ID (SGDID) (mandatory)
2.   Feature type (mandatory)
3.   Feature qualifier (optional)
4.   Feature name (optional)
5.   Standard gene name (optional)
6.   Alias (optional, multiples separated by |)
7.   Parent feature name (optional)
8.   Secondary SGDID (optional, multiples separated by |)
9.   Chromosome (optional)
10.  Start_coordinate (optional)
11.  Stop_coordinate (optional)
12.  Strand (optional)
13.  Genetic position (optional)
14.  Coordinate version (optional)
15.  Sequence version (optional)
16.  Description (optional)
```

Step 7.

Count total genes

Step 8.

Count only verified genes. Count only uncharacterized genes.

Step 9.

What other types of genes are in this file? For this, you may want to use the **sort** command with the **-u** flag, which will sort the input alphabetically, then take only unique lines.

Step 10.

From the same directory as your fasta files, see if you can predict what each of these commands will do (then try it)
a) head *
b) head *.fsa
c) head chr1*.fsa
d) head chr1*
e) head chr*1.fsa
f) head chr*1
g) grep 'S288C' *
h) grep 'S288C' *.fsa
i) grep 'BK006935.2' *
j) cat * | grep 'BK006935.2' (what's the difference in the output between this one and the last one?)
k) head *.fsa | grep 'chr'
l) head *.fsa | grep 'chromosome' (what's the difference in the output between this one and the last one?)

Step 11.

Building a pipeline

```
$ wget ftp://ftp.imicrobe.us/abe487/yeast/palinsreg.txt
```

a) These are detected terminator sequences in the *E. coli* genome (using the program GeSTer, if you're curious).
b) The command **grep '/G=[^]*' somefile** will find all lines that match /G=*somegenename*, where somegenename is a sequence of non-blank characters. Read the output of **man grep** and figure out how to -*o*nly print /G=*somegenename*, rather than the whole line.
c) Pipe the results of part b) through a **cut** command to get only everything after the =
d) Store the results of part c) in a file named "terminated_genes.txt'
e) BONUS: google for a Unix command that only keeps each gene once, rather than once per annotated terminator.

Step 12.