# Laboratory 4: LINUX command–line text processing tools

LINUX excels at the manipulation of text files. There are many different utilities for dissecting and quantifying files that use either relative position or some sort of matching function. This laboratory focuses on the most commonly used text utilities. Before proceeding with the exercises please read the man pages for the following programs:

`awk` == a pattern scanning and text processing language

`cat` == concatenate files

`diff` == find differences between two files line–by–line

`grep` == identify lines using exact matching (globally search a regular expression and print)

`head` == output the first part of a file

`join` == join lines of two files using a common field

`perl` == practical extraction and reporting language[1]

`sed` == stream editor for filtering and transforming files

`sort` == sort lines of files

`split` == split a file into pieces

`tail` == output the last part of a file

`tr` == translate/transliterate (or delete) characters

`tre-agrep` == identify lines using approximate matching[2]

`uniq` == unique (or not) lines

`wc` == word (and other things) count

### Data description

In this lab, you will analyze data from a study of human gut microbiota. Dethlefsen et al. (2008)[3] used 454 pyrosequencing of 16S rDNA to estimate gut microbiota species composition and abundance in three healthy humans—'A', 'B', and 'C'. The guts of the humans were surveyed before and after a short course of the antibiotic ciprofloxacin. The composition of the gut microbiota changed radically in response to the antibiotic, but after four months the microbiota population had (mostly) returned to the state observed before ciprofloxacin treatment.

[1] A bacronym. [2] Must be installed before you can read the man page (type `sudo apt install tre-agrep` in the terminal). [3] https://doi.org/10.1371/journal.pbio.0060280

## Obtaining and transforming the data

(1) Download supplemental table 3 by typing `wget https://doi.org/10.1371/journal.pbio.0060280.sd003` in the terminal.

(2) Download supplemental table 5 by typing `wget https://doi.org/10.1371/journal.pbio.0060280.sd005` in the terminal.

(3) Confirm that the files are indeed Microsoft Excel format by typing `file journal.pbio.0060280.sd00*` in the terminal. Answer question (1).

(4) Using the 'mv' utility, rename the files to 'table3.xls' and 'table5.xls', respectively.

(5) Have a look at the .xls files that you downloaded using your favorite spreadsheet program (e.g. gnumeric [type `sudo apt install gnumeric` in the terminal to install]).

(6) To work with the files using command–line tools, they first need to be converted to ordinary text. You could use the 'Save as…' functionality of a spreadsheet program or the command–line utility 'xls2csv'. For the command–version:

   (a) Install the package 'catdoc' which contains 'xls2csv' by typing `sudo apt install catdoc` in the terminal (enter your password if prompted).

   (b) Read the 'xls2csv' man page.

   (c) Convert the files to Comma Separated Variable (.csv) format by typing `xls2csv -q 0 table3.xls > table3.csv` and `xls2csv -q 0 table5.xls > table5.csv` in the terminal.

   (d) Confirm that the converted Excel files have the correct number of lines by typing `wc -l table*.csv` in the terminal. Compare the result to the number of rows in your favorite spreadsheet program.

## Tasks

(1) Make a list of fields for future reference by typing `head -n 1 table3.csv | tr ',' '\n' > table3-list.txt` and `head -n 1 table5.csv | tr ',' '\n' > table5-list.txt` in the terminal. Answer question (2).

(2) Determine which column of table 3 lists family by typing `grep -n Family table3-list.txt` in the terminal. Answer question (3).

(3) Determine the numbers of families (column 24) of microbes sampled in the survey by typing `tail -n +2 table3.csv | awk -F, '{if(length($24)>2){print $24}}' | sort -u | wc -l` in the terminal. Answer question (4).

(4) Determine the number of OTUs sampled from each family by typing `tail -n +2 table3.csv | awk -F, '{if(length($24)>2){print $24}}' | sort | uniq -c > families.txt` in the terminal. View the results in 'families.txt' using 'cat', 'less', 'nano', 'gedit', or some other program.

(5) Count the number of families with only one OTU by typing `awk '{if($1==1){print $1,$2}}' families.txt | wc -l` in the terminal. Answer question (5).

(6) One of the single OTU families is Corynebacteriaceae. To determine if this identification could be an error, one first must determine how similar other 16S tags are to the one that was used to identify Corynebacteriaceae.

   (a) Extract the Corynebacteriaceae 16S tag by first locating the 'Dominant Tag in refOTU' field using 'grep'.

   (b) Use 'grep' and 'awk' to extract the tag into a file called Corynebacteriaceae.txt

   (c) Look for similar sequences by typing `echo -n "tre-agrep -D 1 -I 1 -S 1 -E 10 -s '" | cat - Corynebacteriaceae.txt | perl -pe "{s/\n/\' table3.csv/}" | bash` in the terminal. Answer question (6).

   (d) Look for progressively more dissimilar sequences by changing the '-E' value. Answer question (7).

(7) The family with the most OTUs is Lachnospiraceae. Lets focus on the data from just this family.

   (a) Make a file of Lachnospiraceae data using using grep by typing `grep Lachnospiraceae table3.csv > Lachnospiraceae.csv` in the terminal.

   (b) To find the number of 16S tags that are identical to those found in the reference database, first locate the 'Distance' column using 'grep'.

   (c) Count the number of '0' distance Lachnospiraceae entries by typing `awk -F, '{if($27==0) {print $0}}' Lachnospiraceae.csv | wc -l` in the terminal.

   (d) Count the total number of Lachnospiraceae entries using 'wc' or 'grep' and calculate the proportion that are identical to the reference tags.

   (e) The 'BEGIN' and 'END' statements of 'awk' provide a more efficient way of counting things as illustrated by the calculation of the average distance between tags and references: type `awk -F, 'BEGIN{x=0; y=0}{x+=$27; y++}END{print x/y}' Lachnospiraceae.csv` in the terminal.

   (f) Now calculate the average distance from the reference for everything but Lachnospiraceae by typing `tail -n +2 table3.csv | grep -v Lachnospiraceae | awk -F, 'BEGIN{x=0; y=0}{x+=$27; y++}END{print x/y}'` in the terminal. Is the number substantially different? Answer question (8).

(8) The 'join' utility can be used to merge tables 3 and 5. Before the tables can be merged, they must be sorted and the the fields used for the join must be changed to the same format.

   (a) Remove the field list, sort table 3 on the 'refOTU designation', and remove blank lines by typing `tail -n +2 table3.csv | sort -t, -k 1b,1 | awk -F, '{if(length($0)>2){ print $0}}' > table3-sort.csv` in the terminal.

   (b) Transform the 'refOTU designation' field in table 5 to match that of table 3, remove the field list, remove the first column (it is not needed later), and sort by typing `tail -n +2 table5.csv | tr '_' ',' | awk -F, -v OFS=',' '{if(length($0)>2){print $2"_"$4"_"$3 "_"$5,$6,$7,$8}}' | sort -t, -k 1b,1 > table5-sort.csv` in the terminal. Answer question (9).

   (c) Type `join -1 1 -2 1 -t, -e EMPTY -a 1 table3-sort.csv table5-sort.csv > table35.csv` in the terminal. Answer question (10).

(d) There were 528 OTUs that showed significant differences in abundance between human test subjects. Those OTUs are indicated by '*', '**', or '***' in column 31 of table35.csv (column 5 in table5.csv). Use these entries and 'grep' to confirm that the join worked properly.

(e) Determine the family distribution of those OTUs that have significant changes in abundance by typing `grep '*$' table35.csv | awk -F, '{print $24}' | sort | uniq -c | awk '{print $2,$1}'` in the terminal. Answer question (11).

## Questions (https://forms.gle/BU1NjzGsa68RKEW59)

(1) What type of computer operating system was used to create the supplemental files?

(2) For task (1), explain what 'head' and the 'tr' steps are doing.

(3) For task (2):

(a) Explain what the '-n' option does.

(b) Does grep start counting from zero or one?

(c) How could you complete task (2) without making an intermediate file (task (1))?

(4) For task (3):

(a) Explain each step in the command string.

(b) How would you modify the command string to count the number of genera?

(5) For task (5):

(a) Explain why '==' is used and not '='.

(b) How would you modify the command string to count the number of families with 10–50 OTUs?

(c) Is 'wc' needed?

(d) Could you have used 'awk' instead of 'wc'? How?

(6) For task (6)(c): explain each step in the command string.

(7) For task (6)(c):

(a) What family had the most similar tag to that of Corynebacteriaceae?

(b) How many differences where there?

(c) Would your answer change if you used different insertion/substitution/deletion costs?

(d) Based on this data, do you think the identification was an error?

(8) For task (7):

(a) Could you have done this task without making a separate file for the Lachnospiraceae?

(b) What command string would you have used?

(c) If you wished to sum the total number of Lachnospiraceae occurrences (columns 2–19) in the data set, what command string could you use?

(9) For task (8)(b), explain each step in the command string.

(10) For task (8)(c), explain each step in the command string.

(11) For task (8)(e), explain each step in the command string.

## Literature cited

**Dethlefsen, L., S. Huse, M. L. Sogin & D. A. Relman**. 2008. The pervasive effects of an antibiotic on the human gut microbiota, as revealed by deep 16S rRNA sequencing. PLoS Biology 6: e280.

*Due at the start of class February 21.*