COMMAND LINE COMMANDS

June 24, 2025

Contents

1 cut

For extracting sections of lines from input to stdout

1.1 Useful flags

`cut -f`

→ Field numbers (use with -d)

`cut -d`

→ Cut delimiter (default: TAB)

1.2 Example uses

```
cut -d',' -f2 data.csv
```

→ Extracts second column from data.csv

```
cut -d',' -f1,3 data.csv
```

→ Extracts column 1 and 3 from data.csv

```
cut -d',' -f2-4 data.csv
```

→ Extracts columns 2-4 of data.csv

```
echo "abcdef" | cut -c2-4
```

→ Outputs `bcd`

2 datamash

For quick stats from CLI

2.1 Syntax

`datamash [OPTIONS] operation column`

2.2 Useful operations

`count`

→ Number of rows

`sum`

→ Sum of numeric values

`mean`

→ Average

`min/max`

→ Minimum/Maximum

`uniq`

→ Unique values (sort first)

`groupby`

→ Group rows by a column (use with sort)

2.3 Useful flags

`-t <DELIM>`

→ Set field delimiter (e.g -t',' to separate by commas for .csv)

2.4 Example uses

```
datamash -t',' sum 2 < data.csv
```

→ Sums 2nd column

```
#Compute mean for 1st column in .txt
cat gene_counts.txt | datamash  mean 1
30.785308441558
```

→ Compute mean for 1st column in .txt file

3 grep

For finding occurances within files

3.1 Useful flags

1. `grep -w` to only get exact occurance/whole word instead of part of another.
2. `grep -c` to print number of occurances (line counts) to stdout instead of the actual occurances
3. `grep -A NUM` for `NUM` count of lines after grep target to stdout

3.2 Example uses

```
$ cat vombaticus.gff3 | cut -f 3 | grep tRN
```

```
tRNA
tRNA
tRNA
tRNA
tRNA
tRNA
tRNA
tRNA
```

```
$ cat vombaticus.gff3 | cut -f 3 | grep -c tRN
```

```
8
```

```
$ grep -A1 ">" file.fa
```

To find header of new fasta when separated by >

4 sort-uniq-count-rank

For organising counts in datasets - Essentially runs `sort | uniq -c | sort -rn`

4.1 Example uses

```
cat vombaticus.gff3 | cut -f 3 | sort-uniq-count-rank
```

```
`386346 exon`
`364707 CDS`
`46571  biological_region`
`40512  ###`
`32886  mRNA`
`29973  five_prime_UTR`
`21114  gene`
`20531  three_prime_UTR`
`15416  region`
`3318   ncRNA_gene`
`1911   lnc_RNA`
`700    snRNA`
`577    pseudogene`
`577    pseudogenic_transcript`
`543    miRNA`
`462    transcript`
`311    snoRNA`
`22 tRNA`
`19 scRNA`
`13 rRNA`
`6  Y_RNA`
`2  V_gene_segment`
`1  #!genebuild-last-updated 2019-02`
`1  #!genome-build  bare-nosed_wombat_genome_assembly`
`1  #!genome-build-accession GCA_900497805.2`
`1  #!genome-date 2018-12`
`1  #!genome-version bare-nosed_wombat_genome_assembly`
`1  ##gff-version 3`
`1  J_gene_segment`
```

5 tr

For 'trapping' stdin. Takes input and translates, squeezes, or deletes characters. Use with `echo`, pipes, or redirections.

5.1 Syntax

```
tr [OPTION] SET1 [SET2]
```

5.2 Useful flags

`(none)`

→ Translate SET1 to SET2

`tr -d`

→ Delete characters in SET1

`tr -s`

→ Squeeze repeated characters into one

`tr -c`

→ Complement SET1 (ie match everything else)

5.3 Example uses

```
echo "hello" | tr 'a-z' 'A-Z'
```

→ `HELLO`

```
`tr "\n" " "
```

→ Converts newlines to spaces

```
echo "hello 123" | tr -d '0-9'
```

→ `hello` (deletes characters - numbers here)

```
tr -d '\n' < file.txt
```

→ Removes newlines (flattens text)

6 wc

For printing wordcount, newline count, and byte counts of file

6.1 Useful flags

`wc -l`

→ Line counts

`wc -w`

→ Word counts

`wc -c`

→ Byte counts

`wc -m`

→ Character counts (UTF-8)

6.2 Example uses

```
wc -l file.txt
```

→ Outputs line count for file.txt

```
cat file.txt | wc -l
```

→ Pipe into word count (same output as above)

```
grep "apple" file.txt | wc -l
```

→ Counts how many lines contain `apple`

7 wget

For recursive downloading files from the internet