|  |  |
| --- | --- |
| **Informatics – Advanced** | |
| **Seungchan Oh in BA1** | |
| **Exercise 12.4** | |
| wget http://users.ugent.be/~wmdeneve/informatics/hello.txt | download hello.txt |
| find ~seuoh ! -newer hello.txt ! -samefile hello.txt | find files which is not newer(older or at same time) than hello.txt and not same(different). |
| rmdir ~seuoh/\* 2> /dev/null ; man touch | remove directories including subdirectories which are empty, directories that somethings are inside result as errors and these errors are processed not to be expressed by /dev/null. After that, manual of touch comes out. |
| find ~seuoh -type f -size +50k -exec gzip {} ';' | find files which size are over 50kb and compress them. |
| find ~seuoh -name \*.gz -ok gunzip {} ‘;’  Y | find .gz files, and unzip them with permission |
| find . -type d ! -path ‘./personal/\*’ | find directories which is not in personal directory |
| find /etc -type f -name ‘^[A-Z].\*[0-9]$’ | find files in etc directory which name starts with capital word and ends with digit. |
|  | |
| **Exercise 13.5** | |
| ls -l | cut -d”.” -f2 | grep -ve “ ” | sort | uniq -c | sort -r | extract only extensions from list and count how many extensions are. sort the result in reverse order. |
| ls -1 | grep -e “\w\.\w” | cut -d”.” -f2 | sort | uniq -c | sort -r | different way. |
|  | |
| **Exercise 13.6** | |
| wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/  HG00096/sequence\_read/SRR062635.filt.fastq.gz | download the file. |
| wc -l SRR062635.filt.fastq | count how many lines in the fastq file. |
| sed -n -e ‘1,100p’ SRR062635.filt.fasq > srr.fasta | show only lines from 1st to 100th and save as srr.fasta |
| sed -n '1~4s/^@/>/p;2~4p' srr.fastq > srr.fasta | From every 1st to 4th lines, replace starting ‘@’ to ‘>’ and, append 2nd line in every 4 lines. |
|  | |
| **Exercise 14.3** | |
| gunzip coding\_regions.txt.gz | unzip the file. |
| sed ‘s/\(…\)\(…\)/\1 \2 /g’ coding\_regions.txt > codons.txt | put interval to the text with every 3 words and save output as codons.txt |
| egrep -c ‘UGA|UAA|UAG’ codons.txt | count lines which contain any stop codons. |
| grep -cv ‘\([^ ][^ ][^ ]\).\*\1’ codons.txt | count lines which one codon appears again |
| grep ‘GU[AUCG]’ codons.txt | grep -c ‘CC[AUCG]’ | find lines which contain valine first and count lines which contain proline from output. |
| egrep ‘CG[AUCG]|AG[AG]’ codons.txt | egrep -cv ‘UC[UCAG]|AG[UC]’ | find lines which contain arginine and count lines which does not contain serine from output |
| grep -cv ‘GU[^ ].\*GU[^ ]’ codons.txt | count lines which doesn’t contain valine twice and more. |
|  | |
| **Exercise 14.4** | |
| 1,$s/[ \n] \*/ /g  1,$s/}, /},^M/g  1s/{ //  $s/ } // | shrimp all margin including line feed to only one space.  give a line feed after characteristics of each element.  delete starting and end curly bracket of this text. |
| 1,$s/{\* ”[ A-Za-z0-9\_(\/)]\*” ://g  1,$s/ [ }],\*/ /g | remove all properties of elements which pattern is ‘that :’ is next to each property, |
| 1,$s/”\*, ”\*/;/g  1,$s/”\(.\*\)” : “\(.\*\)” $/\1;\2 /g | replace all commas to semicolons which are located between values.  replace all colons after name of elements to semicolons and delete spaces. |
| 1,$s/\n.\*[-a] $// | remove all values which are not numerical at the end of each line. |
| 1,$!sort -t’;’ -k16,16nr -k1,1 | ! is such a pipeline connection and sort with option t, which makes a standard ‘;’. sort with 16th order in reverse of numerical ascending order. And then, sort with first order as standard. |
|  | |
| **Exercise 15.1** | |
| unzip periodic\_table.zip | unzip files |
| vim periodic\_table\_data.csv | open the data file with vim |
| Q | Ex mode |
| 1,$s/\([0-9]\*\), \([A-Za-z]\*\) , \([A-Za-z]\*\) \*,\(.\*\)/\1,\2,\3,\4/ | substitute lines with spaces to ones without them. |
| 1,$s/Ancient$/0/ | substitute ‘Ancient’ at last order to 0. |
| :wq periodic\_table\_data\_cleaned.csv | save as other name and quit |
| vim periodic\_table\_history.csv | open the history file with vim |
| Q | Ex mode |
| 1,$s/\([0-9]\*\), \([A-Za-z]\*\) , \([A-Za-z]\*\) \*,\(.\*\)/\1,\2,\3,\4/ | substitute lines with spaces to ones without them. |
| 1,$s/,Ancient,/,0,/ | substitute ‘Ancient’ between commas to 0. |
| :wq periodic\_table\_history\_cleaned.csv | save as other name and quit. |