some useful bash commands

command -option(s) argument(s)

- **man** : interface to system reference manuals
- **who** : show who is logged on
- **pwd** : present working directory
- **ls** : list directory contents
  - **ls -l *filename*** : list detailed information on file
- **wc *filename*** : list line, word, and byte count of file
- **head *filename*** : return first 10 lines of file
- **tail *filename*** : return last 10 lines of file
  - **tail -n +2 *filename*** : return all lines of a file except the first line
- **chmod ooo *filename*** : change read/write/execute mode, where ooo is octal values for owner/group/all
  - octal values = the sum of 4 (read), 2 (write), and 1 (execute)
- **set -o noclobber** : set session variable to prevent accidental overwriting of files. appends still allowed.
- **cat -vet *filename*** : read file contents, displaying hidden characters.
- **less *filename*** : display file contents one screen full at a time, press SPACEBAR for next screen, **p** for previous screen, **q** for quit
- **cp *source destination*** : copy source file to destination
- **mv *source destination*** : move source file to destination, also used to rename files
- **mv -v *filename-substring destination*** : move files with same substring in name; use with wildcard *
- **sort -k*n* -n *filename*** : sort contents of file
  - where -k means *column*
  - where -n means *numeric sort*
  - where -c means *count*
  - where -n means *numeric sort*
- **cut -c*n* *filename*** : cut column *n* from file
- **egrep *string filename*** : search for string in file
- **top** : display LINUX processes
- **ps *aux*** : list processes in memory, all info
- **kill *PID*** : kill process by process ID
- **time *statement* or *script*** : return how long a given command or script takes to run
- **watch *statement*** : re-run a statement at a regular interval, & update output

regular expressions

filename = */usr/share/dict/american-english*

- the metacharacters
  - { } : repetition modifier ex: am means exactly m a's; am, means at least m a's
  - [ ] : class of characters ex: [AEIOUaeiou] means match any vowel
  - ( ) : group & remember this group for reference later
  - ^ $: anchor to beginning of string/anchor to end of string
  - . : any character except newline
  - | : alternative, match exactly one of the alternatives
  - * : match zero or more times
  - + : match one or more times
  - ? : match zero or one times
  - \: escape – makes the following metacharacter into a literal, or makes character into a metacharacter
- examples
  - egrep **"^ a(.)?e$"** filename *matches* ace age ale
  - egrep **"^ ape[rx].*$"** filename *matches* aperitif apertures apex's
  - egrep **"[aiou](e)+[aiou]$"** filename *matches* Gaea diarrhoea milieu
  - egrep **"(re){ 2, } $"** filename *matches* Nyerere
- how to return count of all matches per line
  - egrep -o -n "pattern" *filename* | cut -d : -f1 | uniq -c
  - where -o means *print only matched part of line*
  - where -n means *prefix output with line number*

agrep: find matches with indels & subs

- the options: -n -Ic -Dc -Sc , -n (currency, max. 6), -Ic (cost for each insertion), -Dc (cost for each deletion) ; -Sc (cost for each substitution)

Bash standard I/O

- 0 : STDIN : standard IN (keyboard)
- 1 : STOUT : standard OUT (terminal)
- 2 : STDERR : standard ERROR (terminal)

Just Enough: Bash Cheat Sheet

Introduction to Programming for Researchers

Bash Command Line Shortcuts

GNU readline library

- CTRL-a : go to start of line
- CTRL-e : go to end of line
- ALT-f : go forward one word at a time
- ALT-b : go back one word at a time
- CTRL-k : cut from cursor to end of line
- TAB-completion
  - single TAB: completes name of cmd, file, or dir
  - double TAB: lists all cmds, files, or dirs whose names start with string
- UP-arrow : scroll up one line in bash history
- DOWN-arrow : scroll down one line in bash history
  - RETURN : execute the command
  - CTRL-u RETURN : return to blank command line
- CTRL-r : REVERSE SEARCH enter *string*; returns first matching line in reverse order
  - RETURN executes matching line
  - CTRL-r: find next matching line
  - CTRL-g: abandon search and return to original line
- **history** : returns enumerated history of commands entered on command line
  - !n : execute line number n
  - history | egrep *string* : returns all lines in history containing *string*

Bash operators

- **> *filename*** : write output to *filename*
  - **2> */dev/null*** : redirect errors to the bit bucket
- **> > *filename*** : append output to *filename*
- ***cmd1* | *cmd2*** : pipe output of *cmd1* as input to *cmd2*
- ***cmd1* && *cmd2*** : execute *cmd2* iff *cmd1* executes
- ***cmd1* || *cmd2*** : execute *cmd2* iff *cmd1* fails

gawk: pattern scanning language

- **gawk built-in variables RS**=record separator, **FS**=field separator, **OFS**=output field separator, **FILENAME**, **NR**=number of records processed
- **gawk** 'BEGIN { FS = "*sep*" } ; { print $*column* }'
- **gawk** 'BEGIN { FS = "*sep*" } ; *regex*\{ print }'
- to print contents of all files in a directory, separating each file content with a space and including the filename:
  gawk 'FNR==1{print ""} FNR==1{var=FILENAME; n=split(var,arry,//\/); print arry[n]}1' *directory*

shuf - generate random permutations

- **shuf -n N *filename*** : using uniform distribution, print N random lines from *filename*. Default is without replacement. -r allows replacement.
- **shuf -i 1-10 | paste -s -d ','** : randomize ints from 1 to 10, inclusive. pipe output to paste all items on one line, separated by commas

sed - stream editor

- **sed -i *s/search_str/replace_str/g* *filename*** : find-and-replace all occurences of search_str with replace_str
- **sed -i 's/\s*$//g'** : remove trailing whitespace in-place
- **sed -i '/^$/d'** : delete empty lines
- **sed 's|$FILE|"$FILE"|g'** : use vertical bars, instead of slashes, when search or replacement string has quotes
- **sed -n *'1,5p'*** : print lines 1 to 5
- **sed -n *'1~2p'*** : start at line 1, print every 2d line
- **sed -n *'/search_str /p'*** : print lines with *search_str*
- **sed *'3d'*** : delete line 3 from output
- `sed -i '/---/! s/—/---/g'`
  if line doesn't have 3 consecutive n-dashes, replace any occurrence of m-dash with 3 consecutive n-dashes in line

Vi IMproved, a programmer's text editor

- **vim** *filename* : create a file with *filename* or open existing file
- **vim** +nnn *filename* : open file at line number nnn
- **vim** +/{*pattern*} *filename* : open file on first line containing *pattern*
- ESC : go into command mode. when in doubt which mode you're in, hit ESC
- *command mode*
  - :syntax on :turn on syntax highlighting
  - :set nu : set line numbering on
  - :nonu : set line numbering off
  - :nnn : go to line number nnn
  - :1 : go to line 1 in file
  - :$: go to last line in file
  - $: go to end of current line
  - d$: delete from cursor position to end of line
  - ^ : go to start of current line
  - x : delete character under cursor
  - d d : delete line under cursor
  - d w : delete word under cursor
  - y y : yank (copy) line under cursor
  - p : paste
  - u : undo
  - :wq : write file to disk and quit vim
  - :w! : write file to disk, stay in vim
  - :q! : abandon changes to file, quit vim
  - i : enter editing mode, insert text at cursor
  - a : enter editing mode, append text at cursor

tmux - terminal multiplexer

- **CTRL b** : default prefix
- **tmux new** : create tmux session (for single session)
- **CTRL b** d : detach single session
- **tmux attach** : attach single session

- **tr -d '[:punct:]' < "$FILE"** : read in contents of $FILE to **tr**, which will delete all punctuation from the stream
- **tr -s '\r' '\n' < "$DOSFILE" > "$LINUXFILE"** : replace all CLRF EOL characters with \n characters, and write out result to $LINUXFILE.

ipython

GNU readline library

- % : the prefix for ipython line magic
- %% : the prefix for ipython cell magic
- to set up ipython so that an edited, imported module is automatically reloaded into session memory:
  - %load_ext autoreload
  - autoreload 2
- CTRL-q CTRL-j : insert newline in multiline input
- %hist -g *searchstring*: search all sessions of ipython for lines containing *searchstring*
- %save *filename* nnn-ppp : save lines nnn through ppp of current session to *filename*.py
- %edit *filename* : open *filename* in ipython text editor. default text editor is vi/vim.
- %whos : show all objects in session memory
- %run -p *filename* : run and profile *filename*
- %rerun *nnn* : rerun command in line number *nnn*
- %reset : reset ipython session

tr - translate or delete characters

- **arguments** :
  - -c : use the complement of SET
  - -d : delete characters in SET
- **escape sequences** :
  - \b : backspace
  - \f : form feed
  - \n : new line
  - \r : return
  - \t : horizontal tab
  - \v : vertical tab
- **sets** :
  - [:alnum:] : all letters and digits
  - [:alpha:] : all letters
  - [:blank:] : all horizontal whitespace
  - [:cntrl:] : all control chars
  - [:digit:] : all digits
  - [:graph:] : all printable chars, not including space
  - [:lower:] : all lower case letters
  - [:print:] : all printable chars, including space
  - [:punct:] : all punctuation chars
  - [:space:] : all horizontal or vertical whitespace
  - [:upper:] : all upper case letters