

# III. Monitoring process & Shell script

Wei, Fu-Jin

July 9<sup>th</sup>, 2019

2019 GS summer camp

# Shell script

- A kind of computer program language. Like the macro in SAS or R, to control other program or process to work.
- Main part in my idea
  - Variable : store values and information
    - `A=${B}`
  - Conditional statement : pass or suspend, specific process or don't mind
    - `If... ; then... ; fi`
  - Loop control : similar job, but multiple time
    - `for...; do... ; done`

# Customized Variables

- Other than essential variable, all preset variable wrote in shell preload profiles or scripts, eg: .profile , .bashrc, .bash\_profile
- Temporary environment variable could be defined by “export”
- The variable for this shell only
- Use “{””}” to precisely parse

```
$ noexpV="extremely"  
$ export expV="happy"  
$ echo ${noexpV} ${expV}  
$ env | grep expV  
$ bash  
$ echo ${noexpV} ${expV}  
$ exit
```

# if... then... ; else...; fi (1)

- Basic form

- -f : True if FILE exists and is a regular file.

```
$ if [ -f ~/.ncbirc ]; then echo "Yes" ;  
  else echo "No" ; fi
```

- Other expression for FILES

- -a : True if FILE exists.
- -d : True if FILE exists and is a directory.
- -h : True if FILE exists and is a symbolic link

```
$ b="/usr/bin/blastn"  
$ if [ -f ${b} ]; then echo "regular file";  
  elif [ -d ${b} ] ; then echo "directory" ;  
  else echo "nothing"; fi  
$ b="/usrhome/bin64/NCBI-BLAST"  
$ !-2
```

# if... then... elif... fi (2)

- For value or string
  - Compare two values: -eq, -ne, -lt, -le, -gt or -ge
  - Compare two strings: “==” equal, “!=” not equal,

```
$ a=1;b=2
$ if [ ${a} -eq ${b} ]; then echo "Equal";
  elif [ ${a} -gt ${b} ]; then echo "Greater";
  elif [ ${a} -lt ${b} ]; then echo "Less";
  else echo "What ${a} and ${b} ?"; fi
$ a="xxx"; b="zzz"
$ if [ ${a} == ${b} ]; then echo "same";
  elif [ ${a} != ${b} ]; then echo "diff";
  else echo "What ${a} and ${b} ?"; fi
```

# for... do... done (1)

- for i in <list> ; do <something> ; done

```
$ for i in one two three ; do echo ${i} ; done
```

- for ((initial value; stop condition; step))  
do <something>  
done

```
$ for ((i=1;i<10;i++)); do echo ${i} ; done  
$ for ((i=1;i<10;i++)); do for ((j=1;j<i;j++));  
do echo ${i} ${j} ; done ; done
```

# Get the list and modify the values

- Let file list as the list

```
$ for i in `ls /sharedata/AgBIC_eg/rice_chrom/`;  
do echo ${i} ; done  
$ for i in `ls /sharedata/AgBIC_eg/rice_chrom/*`; do  
echo ${i} ; done
```

- Simply substitute left or right side of string

```
$ x='ex04a/ex04b/ex04c/ex04d'  
$ echo ${x} ${x#*/} ${x##*/}  
$ echo ${x} ${x%/*} ${x%%/*}  
$ for i in `ls /sharedata/AgBIC_eg/rice_chrom/*`; do  
FA=${i##*/} ; CHR=${FA%.fa} ; echo ${i} ${FA} ${CHR};  
done
```

# Make an executable script

- Edit a shell script

\$ vi sleep5.sh

```
#!/bin/bash
Pr=${1:-"1st Run"}
for ((i=0;i<10;i++))
do
    n=`printf %02d ${i}`
    echo ${Pr} ${n} `date +%H:%M:%S`
    sleep 5
done
```

- Make it executable

\$ chmod +x sleep5.sh

- Execute it

\$ ./sleep5.sh



# Semi-manually multiple threads

"&" : run in the background

- Edit a superior controller

```
$ vi runSleep5.bsh  
$ chmod +x runSleep5.bsh  
$ ./runSleep5.bsh
```

```
#!/bin/bash  
for i in 1st 2nd 3rd 4th 5th  
do  
    ./sleep5.bsh "${i} run" &  
done  
echo "All done ?"
```

- Let it run and wait

```
$ vi runSleep5.bsh  
$ ./runSleep5.bsh
```

```
#!/bin/bash  
for i in 1st 2nd 3rd 4th 5th  
do  
    ./sleep5.bsh "${i} run" &  
done  
wait  
echo "All done! For real!"
```

# Monitor the processes

- Those you're running

```
$ ps aux | grep student00
```

```
$ ps auxf | grep student00
```

- Interactive process viewer

```
$ top
```

```
$ top -d 120 -u student00
```

```
$ htop
```

# Compress and backup

- Many files and directors into one file

```
$ cd /sharedata/AgBIC_eg
```

```
$ tar cf ~/ex03/WRKYs_Gmax.tar WRKYs_Gmax
```

- Compress a file

```
$ bzip2 -9 WRKYs_Gmax.tar
```

- Many files and directors into a smaller file

```
$ tar cjf ~/ex03/WRKYs_Gmax2.tar.bz2 WRKYs_Gmax
```

# Execute R via CLI

```
$ tar xjf /sharedata/AgBIC_eg/2019GS.tar.bz2 -c ~/ex03  
$ R -vanilla < test_44k_plot_PCA.r  
$ ls -lrt  
$ R -vanilla < 10fold_pumkin.R
```

# Resources

- Expression used with if
  - [http://tldp.org/LDP/Bash-Beginners-Guide/html/sect\\_07\\_01.html](http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_07_01.html)
- Tutorial of shell script
  - <http://www.tutorialspoint.com/unix/unix-what-is-shell.htm>
  - [http://www.gnu.org/software/bash/manual/html\\_node/index.htm](http://www.gnu.org/software/bash/manual/html_node/index.htm)