

# The BASH Shell

# BASH

---

- the Bourne Again SHell, BASH, is the default shell for Linux
- type `echo $SHELL` to see what shell you're using
- type `bash` to get a bash shell
- Remember `tab` is your friend
- Some basic commands

```
cd
```

```
cd ~bcaffo/ebwg/
```

```
cd ~/ebwg
```

```
cd ~kbroman/
```

```
cd ..
```

```
pwd
```

```
ls --color
```

# Some more examples

---

```
ls -al
more file
less file
grep -i "error" sasfile.log
man grep
lpr file.ps
cp
cp -r
rm
ps -u bcaffo
jobs
kill
wc -l data.txt
cut -c10-15 data.txt
```

The last one extracts columns 10-15 from `data.txt`

# More basics

---

- I/O redirection

```
R --save < sim.R
```

```
R --save < sim.R > sim.log
```

```
R --save < sim.R >> sim.log
```

- Pipelines

```
ls -al *.txt | wc -l
```

```
xlsfonts | grep "courier" | more
```

- Background

```
nice -n 19 R --save < sim.R &> sim.log &
```

- Forgot the &? Use `C-z bg`

- Forgot to nice? Use `renice`

# Some control sequences

---

C-c : stop current command  
C-u : erase command line  
C-\ : really stop current command  
C-z : suspend current command  
C-a : beginning of command line  
C-e : end of command line  
C-k : kill to end of line  
C-p : previous history item  
C-n : next history item  
C-r : search history  
C-l : clear screen  
TAB : completion  
TAB TAB : completion list

**If some of these don't work, you may not be in emacs mode, type**  
set -o emacs

# Startup file

---

- `.bashrc` gets run every time a bash shell is opened
- `source .bashrc` reruns it
- `alias`

```
alias more="less"  
alias Rcs="R CMD SHLIB"  
alias rehash="source ~/.bashrc"
```

- Adding to your path

```
export PATH=$PATH:/home/bcaffo/bin
```

- Printing environment variables

```
echo $PATH
```

# Scripts

---

- A script is a file containing shell programs
- Be sure to change the mode of scripts to execute
- `clean` (does not work on hidden files)

```
rm -f *~  
rm -f *#
```

- `chmod +x clean`
- shell scripts can take arguments
- `bioput`

```
scp $1 biopile.mmi.jhsph.edu:$2
```

# Script example: niceR

---

```
#niceR shortcut for nicing R jobs
#usage: niceR inputRfile outputRfile
#Author: Brian Caffo
# Date: 10/01/03
if [ $# != "2" ]; then
    echo "usage: niceR inputRfile outputfile"
elif [ -e "$2" ]; then
    echo "$2 exists, I won't overwrite"
elif [ ! -e "$1" ]; then
    echo "inputRfile $1 does not exist"
else
    echo "running R on $1"
    nice -n 19 R --no-save < $1 &> $2
fi
```