Unix Tutorial for Biologists

Common Commands
Nathan Dunn
CASSPR (ndunn@cas.uoregon.edu)
Overview

- Unix Editors
- Bash Scripting
- Sample Work Flows
- Other Resources . . .
Unix Editors

• Set cap-locks to controls
• Nano / pico: good to begin with
• Vim
• Emacs
Nano

• nano <filename>

• nano small_sample_1.fq
Vim

- `vim <filename>`
- harder to learn, but very efficient
- You have 3 modes:
  - Command mode: all keystrokes are interpreted as commands (similar to less)
  - Insert mode: most keystrokes are inserted as text (same as nano)
  - Visual mode: helps to visually select some text
Vim Command Mode

- Command Mode:
  - movement is same as less
  - 'b' forward, 'w' back
  - :w (write)
  - :q (quit)
  - :b (buffer)
  - :e (edit)
  - :%s/<pattern>/<replace>/g
Vim Command Mode

- Copy / paste:
  - `yy` = copy line
  - `p` = paste
  - `dd` = delete line
  - `x` = delete char
Vim Command Mode

- Undo / Redo
  - u = undo
  - cntrl-r = redo
Vim Command Mode

- Command mode -> Insert mode:
  - i = insert
  - shift-i = insert before line
  - a = insert after character
  - shift-a = insert after line
  - escape = return to Command mode
Vim Visual Mode

• Command mode -> Visual mode:
  • v = visually select spot
  • shift-v = visually select line
  • navigate as in command-mode
Emacs

• emacs <filename>

• very powerful, harder to learn

• all commands are the control or meta key
Bash Scripting

- `vim create_files.sh`

```
#!/bin/sh

echo "running script"
```

Run it with this

Make file executable

```
$ ls -l create_files.sh
-rw-r--r-- 1 NathanDunn staff 36 May 14 15:42 create_files.sh

$ chmod +x create_files.sh

$ ls -l create_files.sh
-rwxr-xr-x 1 NathanDunn staff 36 May 14 15:42 create_files.sh
```
Bash Scripts

- Edit Script
- Run Script
Bash Scripting

- ./create_files.sh
  - only if chmod +x

```
NathanDunn:unix-biologists% ./create_files.sh
running script
```

- /bin/sh create_files.sh

```
NathanDunn:unix-biologists% /bin/sh create_files.sh
running script
```
Bash Scripting

- conditional: if / then / else

```bash
if [ <condition> ]; then
    // stuff
else
    // other stuff
fi
```
Bash Scripting

- for loops

```bash
for i in <array>
do
    // stuff here
done
```

```bash
while <condition>
do
    // stuff here
done
```
Sample Work Flows

- Sorting Fastq
- Process Fastq Barcodes
- GEO Accessions
- Pipe Separated Data
- Concatenate File Sets
- Remove Duplicate Patterns
Fastq File

- @Header
- Sequence
- +Comment
- Quality

@HWI-ST0747:210:D05GJACXX:3:2308:19482:126261 1:Y:0: CAACTTCCGCGCCCCAGCGCCCGGAACGCGTACTCCTGTA + =1::DB:D@)0)))+2::?3CD:)?0::;@A<5-9?D@>(;@
Sorting FastQ Files

- combine all headers by group
- header (sample 1) @IRIS:7:1:17:394#0/1
  >> 17.fq
- header (sample 2) @IRIS:7:1:17:394#0/2
  >> 17.fq
for line in `cat small_sample_*.fq`
do
  if [[ "$line" == @* ]];
    then
      GROUP=`echo $line | cut -d: -f4 -s` ;
      echo $GROUP
    fi
  echo $line >> output/$GROUP.fq
done
Process Fastq Barcodes

@HWI-ST0747:210:D05GJACXX:3:2308:19482:126261 1:Y:0: CAACTTCCGCGCCCCCAGCGCCCAGCGAACGCGTACTCCTGTA
+
=1:;DB:D@)0))2::?3CD:)?0::;@A<5-9?D>@(;@

• Dump all of barcode CAACT into my.fastq

• grep -P "\@.*\n^CAACT.*\n.*\n" illumina.fastq > my.fastq

  perl-like for new-lines
  starts-with CAACT
  anything then newline

• sed -e 's/^CAACT//' my.fastq > processed.fastq

  • remove leading barcode
GEO Accessions


gunzip GPL4014_family.soft.gz

grep -v ^! GPL4014_family.soft | grep -v \# | grep ENSDARG00000 | cut -f2 | sort -u | grep "^B|^NM" > processed/GPL.txt ;
Pipe Separated Data

Gene Expression

ext2|Prim-25|Prim-25|pectoral fin musculature|xpat superterm|
extl3|Long-pec|Long-pec|pectoral fin musculature|xpat superterm|
gli3|High-pec|High-pec|pectoral fin musculature|xpat superterm|

Phenotype

ext2|Prim-5|Prim-15|pectoral fin bud|hypoplastic|abnormal|pheno entity|superte|
ext2|Prim-25|Prim-25|apical ectodermal ridge pectoral fin bud|aplastic|abnormal|pheno entity|superte|
ext2|Day 4|Day 4|pectoral girdle|decreased size|abnormal|pheno entity|superte|
ext2|Protruding-mouth|Protruding-mouth|pectoral fin|decreased length|abnormal|pheno entity|superte|

- sort by anatomy

- sort -t'|' -k4 phenoGeneFin.txt > output.txt
Concatenate File Sets

• Combine contents of files with same name in 2 directories

```bash
for i in `ls input1/* | cut -f2 -d\/` 
  do 
    cat input1/$i input2/$i > output_htseq/$i 
  done
```
Remove Duplicate Patterns

- Duplicate entries, but order is reversed

ENSDARG000000017007, **nr4a2a**, 9, 5047097, 5051396, ENSDARG00000044532, **nr4a2b**, 6, 12373999, 12378404
ENSDARG00000040177, rgs1, 6, 34300760, 34303440, ENSDARG00000017860, rgs5b, 2, 6590676, 6593943

......

ENSDARG00000044532, **nr4a2b**, 6, 12373999, 12378404, ENSDARG00000017007, **nr4a2a**, 9, 5047097, 5051396

```
cut -d, -f1-5 Duplicates.csv | cat -n | sort -k2 > first_column.csv
cut -d, -f6-10 Duplicates.csv | cat -n | sort -k2 > second_column.csv
# put them together again
cat first_column.csv second_column.csv > combined_columns.csv

# sort by gene name
# do a unique by gene name to get rid of duplicate genes
# sort by first number
sort -k2 combined_columns.csv > sorted_genes.csv
uniq -s7 sorted_genes.csv | sort > unique_genes.csv
```
Remove Duplicate Patterns

- Correct, but in different lines

```bash
# put back together (either order) based on the number
# remove leading numbers
COUNTER='a'
OUTPUT="removed_dupes.csv"
rm -f $OUTPUT
cut -c8-1000 unique_genes.csv | while read i
doi
    if [ "$COUNTER" == a ]; then
        COUNTER='b'
        LINE="$i"
    else
        LINE="$LINE,$i"
        COUNTER='a'
    fi
echo $LINE >> $OUTPUT
done
```

```
vim
%s/^.*\t(.*\n.*\t.*)\n.*\t.*\n\1,\2/
```
Other Resources

• Do it!
• Google it!
• Ask!
• Conery Class: CIS 170
Next Time

• Scripting in perl, python, groovy!