Session #3

Take Home Assignment #2

- Feedback
- Google Form results

awk

- Interpreted programming language
- Great for text processing, data extraction, manipulation of files
- Many one-liners

awk- Printing Columns

- $0 = entire line
- $1 = column 1
- $2 = column 2

```
awk ‘{print $1}’ file
```

awk- Defining Columns

- `awk ‘{print $1}’ file`

Will separate by white space

- `awk –F “\t” ‘{print $1}’ file`

Will separate by tabs

- `awk –F “,” ‘{print $1}’ file`

Will separate by comma

Conditional if Statements

```
If condition x is true:
    do action A
```

```
If column 1 is equal to 5:
    Print line
```

```
If column 1 contains 5:
    Print line
```

```
If column 1 is less than 5:
    Print line
```

Conditional if else Statements

```
If condition x is true:
    do action A
Else:
    do action B
```

```
If column 1 is equal to 5:
    Print line
Else:
    print Not 5
```

```
If column 1 is equal to 5:
    Print column 1
Else:
    print column 2
```

And – Meeting Both Requirements

```
If condition x is true AND condition y is true:
    do action A
Else:
    do action B
```

```
If column 1 equals 5 AND column 3 equals 10:
    print column 1
Else:
    print column 2
```

Or – Meeting At Least One Requirement

```
If condition x is true OR condition y is true:
    do action A
Else:
    do action B
```

```
If column 1 equals 5 OR column 3 equals 10:
    print column 1
Else:
    print column 2
```

awk- Comparison Operators

- `==`
- `!=`
- `~ /pattern/`
- `>`
- `<`
- `>=`
- `<=`

awk- Conditionals

- `awk ‘{if ($1 == 5) print $0}’ file`

If column one is equal to 5, print line

- `awk ‘{if ($1 != 5) print $0}’ file`

If column one is not equal to 5, print line

- `awk ‘{if ($1 > 5) print $0}’ file`

If column one is greater than 5, print line

- `awk ‘{if ($1 ~ /5/) print $0}’ file`

If column contains 5, print line

The GSBS Bootstrappers

awk- Conditionals

- `awk ‘{if ($1 == 5) print $1 ; else print $2}’ file`

If column one is equal to 5, print column 1; else print column 2

awk- Conditionals

-
  ```
  awk ‘{if ($1 == 5 && $3 ==10) print $1 ; else print $2}’ file
  ```

If column one is equal to 5 AND column 3 is equal to 10, print column 1; else print column 2

awk- Conditionals

- `awk ‘{if ($1 == 5 || $3 ==10) print $1 ; else print $2}’ file`

If column one is equal to 5 OR column 3 is equal to 10, print column 1; else print column 2

Exercise Set #5

Exercise Set #5 Solutions

Break

More Fun with awk

- More helpful awk one liners that I have found useful for bioinformatics analysis
- Power of awk is not limited to what is listed here!

awk- Printing the Nth Line

```
awk ‘{if (NR==1000) print $0’ file
```

Print the 1000th line

awk- Printing Every Nth Line

- `awk ‘NR%10==1’ file`

Print every 10th line starting with line 1

If mod 10(line number) equals 1 print line

- `awk ‘{if (NR%10==1) print $0}’ file`

awk- Replacing Characters

- `awk '{gsub(/foo/,”bar”); print}’ File1.txt`

Replace all instances of foo with bar and print out everything to the terminal

awk- Getting the Next Lines

-
```
awk ’{if ($0 ~ /pattern/) print $0; getline; print $0}’ File1.txt
```

Find all instances of ‘pattern’ and return that line as well as the following line.

awk- BEGIN & END Blocks

- `BEGIN {}` – Execute this code first before cycling through each line
- `END {}` – Execute this code after cycling through each line
- `awk ‘BEGIN {First Stuff} {Main Stuff} END {End Stuff}’`

awk- Maximum/Minimum of Column

-
  ```
  awk 'BEGIN {max = 0} {if ($3>max) max=$3} END {print max}’
  ```

Print the maximum value of column 3

-
  ```
  awk 'BEGIN {max = 100} {if ($3 < min) min=$3} END {print min}’
  ```

Print the minimum value of column 3

awk- Sum of a Column

- `awk 'BEGIN {SUM=0} { SUM += $1} END { print SUM }' File1.txt`

Print the sum of column 1

awk- Print a Header

- `awk 'BEGIN {print “Name \t Age”{ print $1 “\t” $2 }' File1.txt`

Print a tab delimited header before the data

awk- Comparing Files

-
  ```
  awk 'FNR==NR {x[$1];next} ($2 in x)’ File1.txt File2.txt
  ```

Store column 1 of file 1 in memory
For each line in file 2, if column 2 is in memory print line

Exercise Set #6

Exercise Set #6 Solutions

The GSBS Bootstrappers

Take Home Assignment #3

You are studying how adding compound X to GM12878 cells changes gene expression. You performed an RNA-seq experiment to measure RNA levels with and without the compound. You had four replicates for each of the conditions and sent off the samples for sequencing.

Your lab's bioinformatician gives you a list of genes with average relative expression levels (CompoundX_DEG.txt). He has already performed the statistical analysis between the samples. The list has all genes (protein coding, ncRNA, pseudogenes) annotated in the GENCODE database. There are 57,445 genes and using excel for your analysis will be impossible!