

## DIFFERENTIAL GENE EXPRESSION ANALYSIS

### Module 6: DESeq2 output parsing

Convert the .csv DESeq2 output file in the DESEQ2 folder to .tsv format:

```
sed 's/,/\t/g' 759_7_from_Scram > 759_7_from_Scram.tsv
```

Let's try viewing it:

```
less 759_7_from_Scram.tsv
```

It has many columns and entries of different lengths, which make it difficult to view. We'll toss in the column command and a couple additional options:

```
column -t 759_7_from_Scram.tsv | less -S
```

That's better. But we probably don't need all these columns. How can we retain the names, log2 fold change, and adjusted p-value in tab-delimited format?

```
awk '{print $2 "\t" $4 "\t" $8}' 759_7_from_Scram.tsv > results.tsv
```

Now, let's filter out genes which we are missing data for.

```
awk '$3 != "NA" {print $0}' results.tsv > results_filtered.tsv
```

It makes sense to order the file with the lowest p-value at the top.

```
sort -k3 results_filtered.tsv
```

This isn't quite what we want, however. It's sorting the values as character strings, not number values. Also, it dropped the header to the very end. Let's avoid this by saving the header line to another file.

```
head -1 results_filtered.tsv > results_filtered_header.tsv  
## head: results_filtered.tsv: No such file or directory
```

Now, we'll do a generalized sort on everything from the second line on:

```
tail -n +2 results_filtered.tsv | sort -g -k3 > \
results_filtered_sorted.tsv

## tail: results_filtered.tsv: No such file or directory
```

Next, we concatenate the new file back on to the header.

```
cat results_filtered_header.tsv results_filtered_sorted.tsv > \
results_filtered_sorted_complete.tsv
```

Pretend we have a list of genes of interest and we want to cross reference it with those in our results. This will be easier if we can compare them directly with the contents of the first row of our file. To do so, we'll remove the quotes.

```
sed -i 's/"//g' results_filtered_sorted_complete.tsv

## sed: -i may not be used with stdin
```

Finally, let's actually do that! Take a guess at what the grep options are doing, then check the documentation with `man` or `-help`.

```
grep -wf /home/elavelle/cancer_genes.tsv results_filtered_sorted_complete.tsv
```