**Differential expression analysis for RNA data**

Example scripts: Kallisto alignment – “Pipeline\_RNAseq\_analysis.slm”

DESeq after kallisto alignment – “DESeq.R”

DESeq after bluebee alignment – “Bluebee\_DESeq2.R”

1. Check what kind of data you have?

All data will need to go through trimming, decontamination, and alignment.

* 1. Is your data single end or paired end reads?
     1. There is a pipeline made for paired end. (This pipeline has the beginnings of code for single end but it is not fully functional.)
  2. What sequencer was used?
     1. The pipeline can be used for most sequencers
     2. If it is Quant-Seq (usually single end) use Bluebee to align, trim, etc.
  3. What did it come from? Human, mouse, bacteria, etc
     1. When choosing pipelines you will need to make sure you have the appropriate reference genome to align to. The pipeline handles mainly human and mouse but can be modified to others. Bluebee has many organism options.
  4. Check all logs after in house alignment pipeline is used!

1. Once aligned, you will need to make sure the output is in count form.
   1. The Bluebee pipeline returns a file called “read\_counts.txt”
   2. Kallisto (used in the pipeline) outputs “abundance.tsv” which can be converted to a count file with the tximport function in R. This will aggregate transcript reads to gene reads.
   3. Other aligners may vary. You might have to research the methods used for those.
2. Once counts are obtained, you will want to normalize the counts. Normalization is required due to the nature of the data.
   1. This can be done with the DESeq package. If you run an initial DESeq process (commented in example code) with defined groups, you can obtain normalized counts.
3. You will also want to do a little more quality control.
   1. Remove genes that have counts of 0 for all samples, or fewer than 5 on average per group.
4. Next, you should run a PCA in R to check the clustering of the data.
5. After the PCA is analyzed, make any outlier or other appropriate changes to data as seen necessary.
6. Run DESeq analysis.
   1. If no changes to data included are made, you can obtain the results from the previous DESeq run.
   2. If multiple comparisons are desired, each comparison can be extracted from the analysis results.
   3. Results are to be in xlsx format and include: ENSG (if applicable), Gene name, Description of gene (optional but helpful), normalized read counts for each sample, log2 Fold Change, Fold Change, p-value, and adjusted p-value

**DESeq program notes:**

1. Make sure the control is specified correctly.
   1. If you only have one comparison, make sure the condition that is listed first is your control. Conditions will be listed alphabetically and if this is to be changed, use the relevel function to specify the reference, or first condition.
   2. If you are doing multiple comparisons, when listing the groups to compare, the first group is the treatment and the second group is the control.
   3. You can double check what the program is considering case and control by looking at the results file; “head({what you label the results file})”
   4. ORDER MATTERS!
2. You can also choose to use the “cookscutoff” option. This is not typically used for small sample sizes or similar conditions (small effect sizes).
3. There is a manual for all of the functions and options of DESeq online.

Further deliverables for clients:

* It is often desired to see the results in a heatmap form. What things are shown on the heatmap may change from client to client.
* There are many heatmap generating programs available in R. Depending on the desired output, some heatmap formats may not work well. It can become a trial and error, and googling expedition for me.
* There is an example of one type of heatmap in the example script.