# Homework Assignment 6
# Total Points: 105
# (Due Thursday November 2, 2023 at 5PM)

Please email your answer (compiled pdf file from R markdown) and R code to Cenxiao (CENXIAO@email.sc .edu).

## Question 1

This is in continuation of Question 2 in Homework Assignment 5. The workflow of microarray data analysis usually follows the steps of (1) reading in data (often from binary files), (2) normalization, (3) differential expression detection and (4) generate report. We will focus on 3 and 4 in this question. We will continue to use the following packages from Bioconductor: oligo (for reading in data and normalization), limma and siggenes (for differential expression), pd.hg.u133.plus.2 (for annotation and generating reports). Refer to Homework 5 to download the microarray data provided through gene expression omnibus under accession number GSE18088. Questions:

1. How many patients developed relapse events? (5 points)
2. In order to identify the differential expression genes between patients with relapse events and patients without relapse, what is the design matrix for this comparison? (15 points)
3. Use limma to detect differentially expressed genes between patients with relapse events and patients without relapse. (15 points)
4. How many genes are differentially expressed under FDR $< 0.05$ in U133Plus2 platform? How many genes with p value $< 0.05$? What are the top 30 differentially genes among them? [Hint: use hgu133plus2SYMBOL to convert Affymetrix probe id to Entrez gene symbols] (15 points)

## Question 2

In this homework, you will need the following data files. Data files:
sampleinfo.txt
GSE60450_Lactation-GenewiseCounts.txt
mouse_c2_v5.rdata
mouse_H_v5.rdata

These files Data files available from: https://figshare.com/s/1d788fd384d33e913a2a You need to download these files and place them in your `/data` directory. The following Bioconductor packages are used for this homework. Packages used:

- limma,
- edgeR,
- gplots,
- org.Mm.eg.db,
- RColorBrewer,

- Glimma

In this homework, we will go through the typical RNA-seq data analysis procedure.

## Overview

- Reading in table of counts
- Filtering lowly expressed genes
- Quality control
- Normalisation for composition bias

## Introduction

Measuring gene expression on a genome-wide scale has become common practice over the last two decades or so, with microarrays predominantly used pre-2008. With the advent of next generation sequencing technology in 2008, an increasing number of scientists use this technology to measure and understand changes in gene expression in often complex systems. As sequencing costs have decreased, using RNA-Seq to simultaneously measure the expression of tens of thousands of genes for multiple samples has never been easier. The cost of these experiments has now moved from generating the data to storing and analysing it.

There are many steps involved in analysing an RNA-Seq experiment. Analysing an RNAseq experiment begins with sequencing reads. These are aligned to a reference genome, then the number of reads mapped to each gene can be counted. This results in a table of counts, which is what we perform statistical analyses on in R. While mapping and counting are important and necessary tasks, today we will be starting from the count data and getting stuck into analysis.

## Data import

*Set up an RStudio project specifying the directory where you have saved the `/data` directory.*

First, let's load all the packages we will need to analyse the data.

```
library(edgeR)
library(limma)
library(Glimma)
library(gplots)
library(org.Mm.eg.db)
library(RColorBrewer)
```

### Mouse mammary gland dataset

The data for this tutorial comes from a Nature Cell Biology paper, *EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival* [@Fu2015]. Both the raw data (sequence reads) and processed data (counts) can be downloaded from Gene Expression Omnibus database (GEO) under accession number GSE60450.

This study examines the expression profiles of basal stem-cell enriched cells (B) and committed luminal cells (L) in the mammary gland of virgin, pregnant and lactating mice. Six groups are present, with one for each combination of cell type and mouse status. Each group contains two biological replicates.

The sampleinfo file contains basic information about the samples that we will need for the analysis today. Detailed sample info can be found in file "GSE60450_series_matrix.txt" from the GEO website. We will also use the counts file as a starting point for our analysis. This data has already been aligned to the mouse genome. The command line tool featureCounts [@Liao2014] was used to count reads mapped to mouse genes from Refseq annotation (see the paper for details).

## Read in and format the count data

1. Read in the sampleinfo and count data file. (5 points)
2. Create an object called "countdata" that contains only the counts for the 12 samples. Store EntrezGeneID as rownames. Shorten the sample names to contain only the first 7 characters. (5 points)

For example, shorten “MCL1.DG_BC2CTUACXX_ACTTGA_L002_R1” to “MCL1.DG” (hint: substr). Assign the shortened names as the column names for the samples.

## Filtering to remove lowly expressed genes

Genes with very low counts across all libraries provide little evidence for differential expression and they interfere with some of the statistical approximations that are used later in the pipeline. They also add to the multiple testing burden when estimating false discovery rates, reducing power to detect differentially expressed genes. These genes should be filtered out prior to further analysis.

There are a few ways to filter out lowly expressed genes. When there are biological replicates in each group, in this case we have a sample size of 2 in each group, we favour filtering on a minimum counts per million threshold present in at least 2 samples. Two represents the smallest sample size for each group in our experiment. In this dataset, we choose to retain genes if they are expressed at a counts-per-million (CPM) above 0.5 in at least two samples.

A CPM of 0.5 is used as it corresponds to a count of 10-15 for the library sizes in this data set. If the count is any smaller, it is considered to be very low, indicating that the associated gene is not expressed in that sample. A requirement for expression in two or more libraries is used as each group contains two replicates. This ensures that a gene will be retained if it is only expressed in one group. Smaller CPM thresholds are usually appropriate for larger libraries. As a general rule, a good threshold can be chosen by identifying the CPM that corresponds to a count of 10, which in this case is about 0.5. You should filter with CPMs rather than filtering on the counts directly, as the latter does not account for differences in library sizes between samples.

3. Use the ‘cpm‘ function from the *edgeR* library [@robinson2010edgeR] to generate the CPM values and then filter. Note that by converting to CPMs we are normalising for the different sequencing depths for each sample. (10 points)
4. How many samples have CPM > 0.5 for the first gene (EntrezgeneID:497097)? (5 points)
5. How many genes have at least two samples with CPM > 0.5? Keep only the genes with at least two samples with CPM > 0.5 for the following analysis. (10 points)
6. For the first sample, plot a scatter plot with the raw count in the y-axis and the corresponding CPM in the x-axis. Add a vertical line at 0.5 CPM and a horizontal line at 10. (5 points)

## Convert counts to DGEList object

7. Create a ‘DGEList‘ object. This is an object used by *edgeR* to store count data. It has a number of slots for storing various parameters about the data. Examine the structure of the newly. created object. What slots are stored in this object? (10 points)
8. Finally, store your final DGEList object as .RData for the next question. (5 points)