BI183 HW #4 – Jennifer Yu

Problem 1

a. NGS Sequencing by Synthesis works by amplification, sequencing, and then analysis. After the DNA has been purified, it gets broken into smaller pieces and given adapters. The next step is the reduced cycle amplification, when sequences for primer binding and terminal sequences are added. This occurs inside of a flow cell, which has oligonucleotides that hold the DNA strands in place while they are being sequenced. Once the fragments have attached, a phase called cluster generation begins. Primers and modified nucleotides enter the chip and force the polymerase to add on only one nucleotide or fluorescent tags one at a time. The polymerases move along a strand of DNA, creating its complementary strand. The DNA strand bends and attaches to the oligo that is complementary to the top adapter sequence. Again, polymerases attach to the reverse strand, and its complementary strand is made. Each strand can attach to an oligonucleotide in the flow cell; this process is called bridge amplification, and it happens for thousands of clusters simultaneously. After each round of synthesis, a camera takes a picture and the wavelength of the fluorescent tag acts as a record. This process continues until all the DNA is sequenced.

b. The MiSeq system costs $99,000 and has output up to 15 Gb with 25 million maximum reads per run in a runtime ranging from 4-55 hours. The maximum read length is 2 x 300 bp. MiSeq is capable of small whole-genome sequencing (microbes, viruses), targeted gene sequencing, gene expression profiling, miRNA and Small RNA analysis and finally DNA-Protein Interaction Analysis, which is the added feature it has over the MiniSeq and iSeq 100. The HiSeq system costs $900,000 and has output from 9 Gb to 1 Tb with 300M – 4B reads per run. The maximum read length is 2 x 250 bp. The HiSeq system offers two run modes and single/dual flow cell sequencing; the dual flow cell has higher output and faster runtime than the single flow cell. The NovaSeq costs $985,000 and generates 6000 Gb with 32-40B reads per run. The maximum read length is 2 x 250 bp. This instrument is a production scale sequencer and it has high quality scores.

c. HiSeq is obsolete and the reagents are very expensive, so there is no pricing on this. For the NovaSeq, the instrument can have up to 40 billion reads in a flow cell per run. As a result, we are assuming that the largest flow cell (S4) is used, which has a reagent kit price of $31,000 for 300 cycles. The library preparation kit is $4,000 for 96 samples. This would work well since the NovaSeq has such high throughput, so large samples would still get good coverage. I am considering the situation where we buy the instrument and library, reagent kits ourselves. Our total cost would be $985,000 for the instrument and $35,000 for 300 cycles, where 40B reads are performed per run (max read length 2 x 250bp). It would take around 4000 reads to get 1 million base pairs, and on each run, we can get 10 million sets of reads that give us 1 million bps each. There are 300 runs per reagent kit, so we would be able to write obtain 3 billion sets of reads with 1 million bps. The total cost with instrumentation and kits is around 1 million dollars, which averages out to around $0.00033 for a million base pairs. It costs $3,400 per run. This is assuming that everything goes smoothly and is maximized. For the MiSeq, the library preparation kit is $1,000 for 24 samples, which is optimal because it improves coverage for the samples. The maximal output reagent kit costs $1869 for 600 cycles. The max read length is 2 x 300 bp and around 25 million reads are performed per run. This means that around 75 sets of

reads with million bps can be read every cycle. There are 600 cycles, so we have 45,000 sets of 1 million bps with each reagent kit. The total cost is the sum of $99,000 for the instrument and $2,869 for the library and reagent kits, which is $101,869. It comes out to around $2.26 to sequence a million base pairs and $169 per run.

Problem 2

a. Describe the different principles:
   i. Manual – This was when only a small number of cells could be manually isolated in single tubes before RNA was extracted via cell lysis. A way to increase the scale was by fluorescence activated cell sorting to isolate cells in microwell plates.
   ii. Multiplexing – This is the practice of processing and analyzing multiple samples at once. In sequencing assays, molecular barcodes are added to cDNA fragments so that material from many cells can be pooled and allows the procedure to be performed in one tube.
   iii. Integrated fluidic circuits – This method works when pressurized liquid flows through the channels in a layer of a chip, and the rubber deflects at the intersection of channels in the bottom layer, creating a valve. The IFCs contain a network of fluid lines, valves and chambers; this makes it possible to perform digital PCR reactions with isolated individual DNA template molecules. It is limited by the set number of chambers used to capture cells.
   iv. Liquid-handling robotics – The principle is to automate the dispensing of reagent, samples, or other liquids to a container. The advance of robotics allowed for a jump to several thousand cells within an experiment.
   v. Nanodroplets - This involves methods to randomly capture and manipulate individual cells in nanoliter droplet emulsions. The inDrop and Drop-seq protocols have ways to isolate cells in droplets and carry out barcoded cDNA preparation. Here, a flow of reagents and beads along with a flow of cells are merged, which is then separated into droplets by the addition of oil. Through calibration, mostly only single cells will be isolated according to Poisson statistics.
   vi. In situ barcoding – This is a strategy where single cells are never individually isolated the lysed, but instead fixed and the mRNA is manipulated in situ inside each cell. Then cells are split into pools of cells, labelled with a barcode, and then pooled and again split randomly into mini pools. This procedure is repeated to amplify the material and create a sequencing library.
b. To perform Drop-seq, first, a single cell suspension needs to be prepared. This is normally done by using robotics to automate procedures and using droplet-based microfluidics to randomly capture single cells into isolated droplets. The droplets are broken open and first-strand cDNA synthesis is performed in bulk. There are a few methods for the next step, second-strand generation. One is to employ a terminal transferase and to add a polyT primer with a PCR sequence, where the cDNA is then amplified by PCR. Alternatively, a template-switching oligonucleotide with a PCR sequence is added and the full-length cDNA can be synthesized by PCR. Finally, it is also possible to use RNase to cut mRNA in the mRNA-DNA duplex. Then, the RNA primed first strand cDNA is used as a template and the second strand is generated by DNA polymerase I. After this, the molecules are sequenced from each end and the reads are first aligned to a reference genome to identify the gene of origin. PCR and IVT methods can amplify

the library. Barcoding can occur during any of these steps. Early methods involved adding cell barcodes during second strand synthesis or during library PCR, where each cell is converted into a single library, and the cell barcodes are equivalent to sample barcodes. More recent methods involve adding cell barcodes during revere transcription and pooling the cells afterwards, so that the downstream procedures are performed in a single reaction. This reduces technical variance between different cells and another level of barcode can be added during library amplification so many samples can be multiplexed and sequenced together.

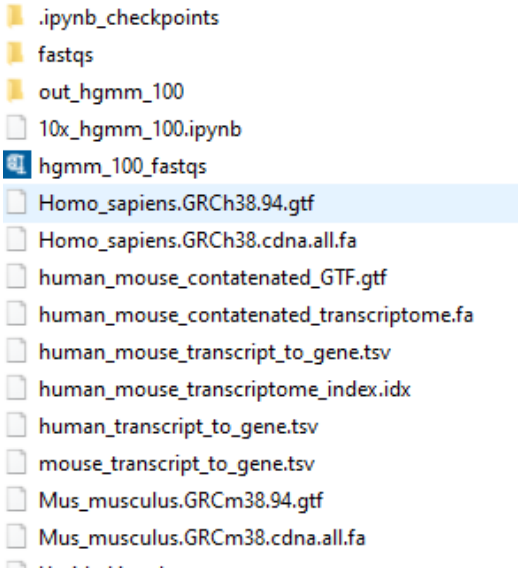
c. Bulk RNA sequencing measures the average expression level for each gene across a population of input cells. This method is good for comparing samples of the same tissue from different species or quantifying expression signatures in studying diseases. However, scRNA-seq has benefits over bulk RNA-seq through its unique properties. It can measure the distribution of expression levels for each gene and study cell-specific changes in transcriptome. The main difference between these bulk and scRNA-seq is that each sequencing library represents a single cell instead of a population of cells. As such, there are more problems with amplification bias and transcript capture efficiency, which are currently being researched. In determining which platform should be used experimentally, it depends on the experiment. One that involves characterizing a specific tissue may do well with a droplet-based method that captures a large number of cells; if studying isoforms, it may be better to undergo full-length transcript quantification.

Problem 3

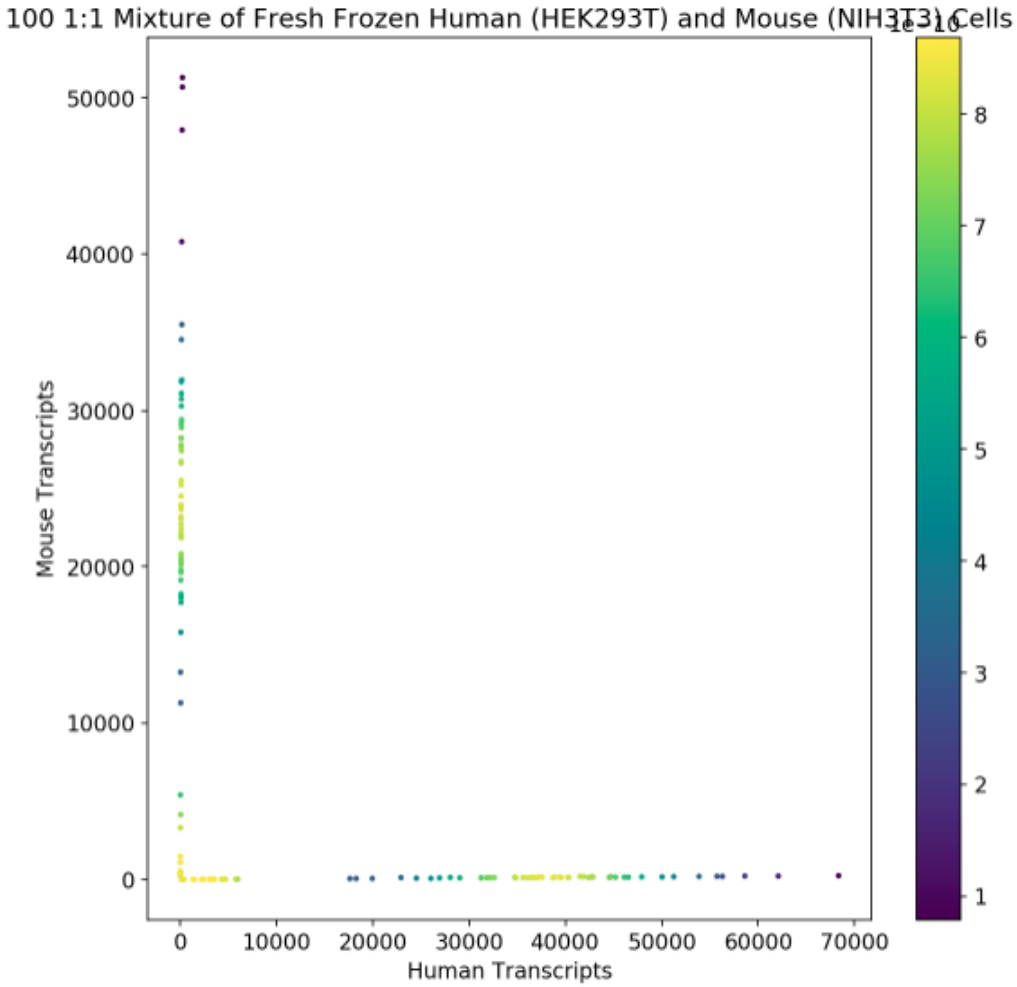
Simpson's paradox is a mathematical inequality that arises when comparing averages with raw data. Here, a trend appears in different groups of data but disappears/reverses when the groups are combined or another variable is introduced. In correlations, two variables may seem to have a positive correlation towards each other, when they actually have negative correlation, due to a "lurking" confounder. This paradox can arise when analyzing single cell data because there may be missing data which is essential for accurate classification. Thus, genome-wide averaging becomes a problem. In the example described in the blog post, original data analysis suggested that exons were more highly methylated than introns, but it was later found that another variable, the quantity of CpGs, was affecting the DNA methylation. We can avoid this by not averaging the data in the tables, discarding rows with missing entries, and impute the missing values. Imputing is a good solution, since discarding data with missing entries means a huge loss of data. There are technologies such as COMPARE which exist to correct plots and replace missing information based on existing data.

Problem 4

a. I ran the notebook. I did most of it manually since the notebook wasn't working very well, so here is a screenshot showing all the files that were downloaded on my machine:



Additionally, here is an image of the barnyard plot generated.



b. The process of mapping equivalent classes (ECs) to genes involves looking for gene differential expression between samples. Equivalence classes in this context refer to the equivalence

relation between two reads, where the equivalence relation (when two reads align to the same set of transcripts) partitions the reads into equivalence classes. One process of mapping ECs is the pseudo alignment used in the kallisto software. Quantification by pseudo alignment begins by using a transcript annotation as reference and then assigning a read as compatible with transcripts that are in close alignment with the read. Reads are, as such, assigned to an EC that reflects the combination of transcripts compatible with it. In kallisto, the comparison of the sequencing reads is done using a de Bruijin graph; it is constructed from k-mers present in the input transcriptome. Each node in the graph represents a k-mer and is associated with transcript(s). Once this is built, then kallisto will store a hash table mapping each k-mer to its contig and its position in the contig (kallisto index). A transcript that contains a node's k-mer is considered to be part of the k-compatibility class of the node. Nodes with the same k-compatibility class are classified as having the same equivalence class. This is how kallisto uses pseudoalignment to map ECs to genes/transcripts.

c. What may help is including some of the potential pitfalls for Windows users with the Anaconda prompt. For example, I was receiving a segfault on bustools, but this had to do with the fact that I was running kallisto bus in Windows without an extra parameter.

Problem 5

I wrote the answers to the questions under the corresponding graphs within the notebook. The HTML version of the R notebook is on the next pages.

The number of genes is as expected for two species. There're way more cells than we expect, which is about 1000. So what's going on?

How many UMIs per barcode?

```
tot_counts <- Matrix::colSums(res_mat)
summary(tot_counts)
```

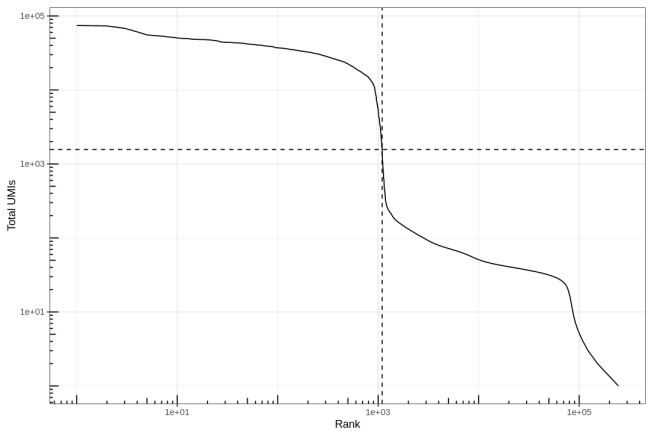
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    1.00    1.00   76.04    8.00 74292.00 
```

The vast majority of "cells" have only a few UMI detected. Using the barcode ranking method, we are going to estimate the amount of empty droplets. There is an inflection point in the data, and we can use this point to see the transition between two components in a distribution.

```
# Compute barcode rank
bc_rank <- barcodeRanks(res_mat)
```

```
#' Knee plot for filtering empty droplets
#'
#' Visualizes the inflection point to filter empty droplets. This function plots
#' different datasets with a different color. Facets can be added after calling
#' this function with `facet_*` functions.
#'
#' @param bc_rank A `DataFrame` output from `DropletUtil::barcodeRanks`.
#' @return A ggplot2 object.
knee_plot <- function(bc_rank) {
  knee_plt <- tibble(rank = bc_rank[["rank"]],
                     total = bc_rank[["total"]]) %>%
    distinct() %>%
    dplyr::filter(total > 0)
  annot <- tibble(inflection = metadata(bc_rank)[["inflection"]],
                  rank_cutoff = max(bc_rank$rank[bc_rank$total > metadata(bc_rank)[["inflection"]]]))
  p <- ggplot(knee_plt, aes(rank, total)) +
    geom_line() +
    geom_hline(aes(yintercept = inflection), data = annot, linetype = 2) +
    geom_vline(aes(xintercept = rank_cutoff), data = annot, linetype = 2) +
    scale_x_log10() +
    scale_y_log10() +
    annotation_logticks() +
    labs(x = "Rank", y = "Total UMIs")
  return(p)
}
```

```
options(repr.plot.width=9, repr.plot.height=6)
knee_plot(bc_rank)
```



The inflection point looks like a reasonable number of cells.

```
# Filter the matrix
res_mat <- res_mat[, tot_counts > metadata(bc_rank)$inflection]
res_mat <- res_mat[Matrix::rowSums(res_mat) > 0,]
dim(res_mat)
```

```
37755 · 1095
```

We get that around 1095 cells are real from the inflection point analysis. This is reasonable because the cell capture rate is claimed to be around 65%, but often can be lower.

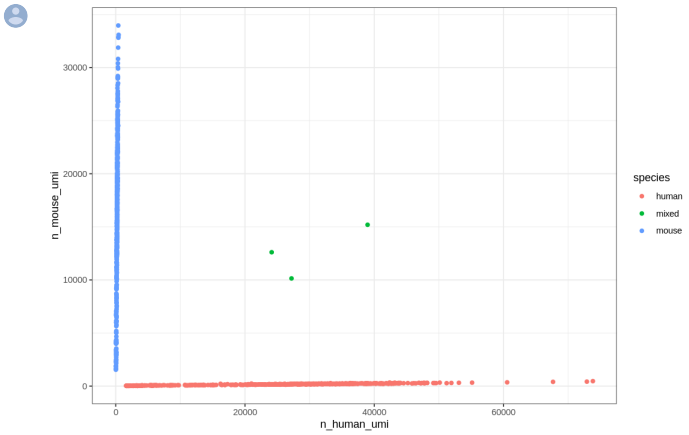
Cell species

How many cells are from humans and how many from mice? The number of cells with mixed species indicates doublet rate.

```
gene_species <- ifelse(str_detect(rownames(res_mat), "^ENSMUSG"), "mouse", "human")
mouse_inds <- gene_species == "mouse"
human_inds <- gene_species == "human"
# mark cells as mouse or human
cell_species <- tibble(n_mouse_umi = Matrix::colSums(res_mat[mouse_inds,]),
                       n_human_umi = Matrix::colSums(res_mat[human_inds,]),
                       tot_umi = Matrix::colSums(res_mat),
                       prop_mouse = n_mouse_umi / tot_umi,
                       prop_human = n_human_umi / tot_umi)

# Classify species based on proportion of UMI, with cutoff of 90%
cell_species <- cell_species %>%
  mutate(species = case_when(
    prop_mouse > 0.9 ~ "mouse",
    prop_human > 0.9 ~ "human",
    TRUE ~ "mixed"
  ))

ggplot(cell_species, aes(n_human_umi, n_mouse_umi, color = species)) +
  geom_point()
```



We can see clearly here that not many of the cells are doublets. The doublets are represented by the green points in the graph, which are much less in quantity than the UMIs for explicitly mice and humans. There's a clear separation between the UMIs for the mice and those for the humans. The cutoff chosen was 90%, which essentially filters out the doublets, which are less than either 90% mouse or human.

```
cell_species %>%
  dplyr::count(species) %>%
  mutate(proportion = n / ncol(res_mat))
```

A tibble: 3 × 3

species	n	proportion
<chr>	<int>	<dbl>
human	566	0.516894977
mixed	3	0.002739726
mouse	526	0.480365297

This data tells us that only around 0.3% of cells are doublets. Note that this is only accounting for mixed cells; it is possible to have a doublet with data from the same species.

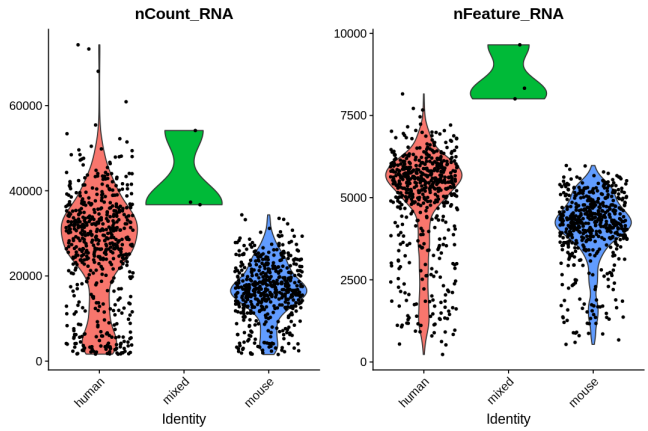
Dimension reduction

```
seu <- CreateSeuratObject(res_mat, min.cells = 3) %>%
  SCTransform(verbose = FALSE)

# Add species to meta data
seu <- AddMetaData(seu, metadata = cell_species$species, col.name = "species")
```

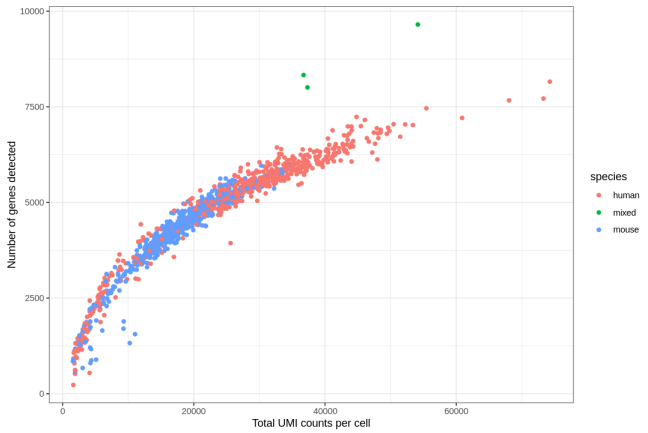
See how number of total counts and number of genes expressed are distributed.

```
VlnPlot(seu, c("nCount_RNA", "nFeature_RNA"), group.by = "species")
```



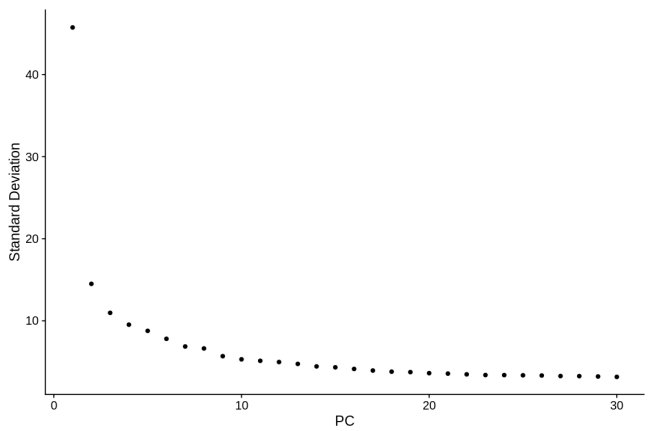
Another QC plot

```
ggplot(seu@meta.data, aes(nCount_RNA, nFeature_RNA, color = species)) +
  geom_point() +
  labs(x = "Total UMI counts per cell", y = "Number of genes detected")
```

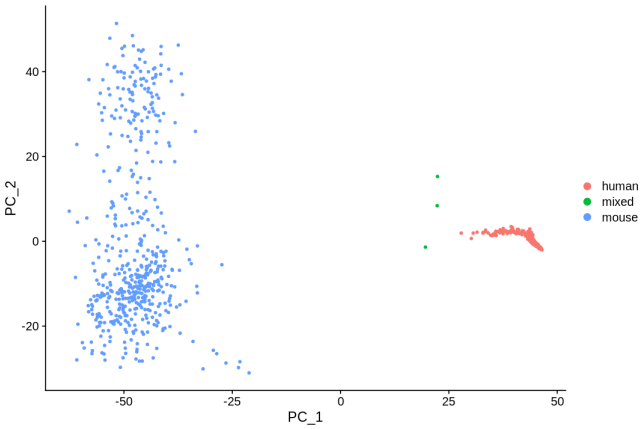


Here, the number of total counts per cell are plotted on the x axis and fraction of human/mouse transcript counts are on the y axis. There is no clean cutoff separating them, but there tends to be more cells classified as human for higher UMI counts per cell. The Internet says that the mouse genome is around 14% shorter than the human genome, which may explain why this plot shows that humans have on average, more genes detected per cell than mice do. If we look above to the plot above this, we can see more visually the distribution of total number of counts and number of genes are distributed.

```
seu <- RunPCA(seu, verbose = FALSE, npcs = 30)
ElbowPlot(seu, ndims = 30)
```

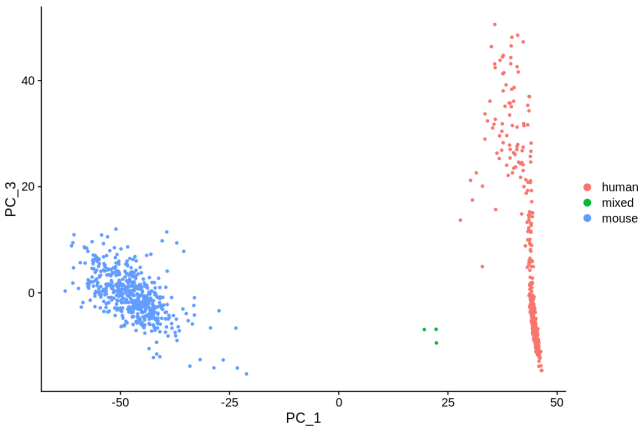



```
PCAPlot(seu, group.by = "species")
```

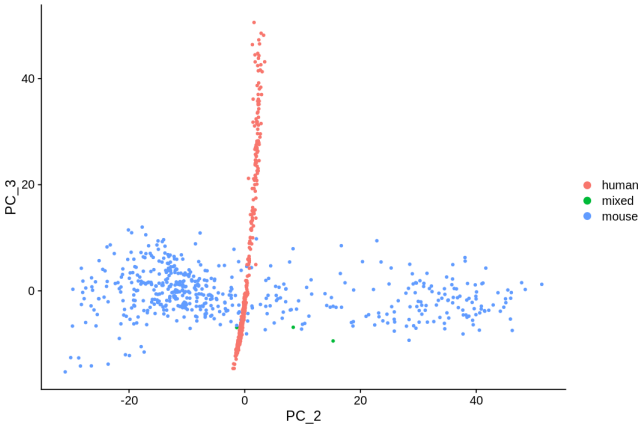


The first PC separates species, as expected. Also as expected, the doublets are in between human and mouse cells in this plot.

```
PCAPlot(seu, group.by = "species", dims=c(1, 3))
```




```
PCAPlot(seu, group.by = "species", dims=c(2, 3))
```



It is important to plot more principal components than just PC1 and PC2 since there may be other sources of variance within the data. In the two PCA plots above, we can see that PC1 vs PC3 and PC2 vs PC3 show differing representations, especially for PC2 vs. PC3, which has the human data within the mass of mouse data. There is another source of variance within the data, though the greatest source is probably shown in the PC1 vs PC2 graph, where the fact that mice and humans are different species contributes to much of variance.