Homework 5

Rohit Kamat rgk359

This homework is due on Feb. 28, 2017 at 7:00pm. Please submit as a PDF file on Canvas.

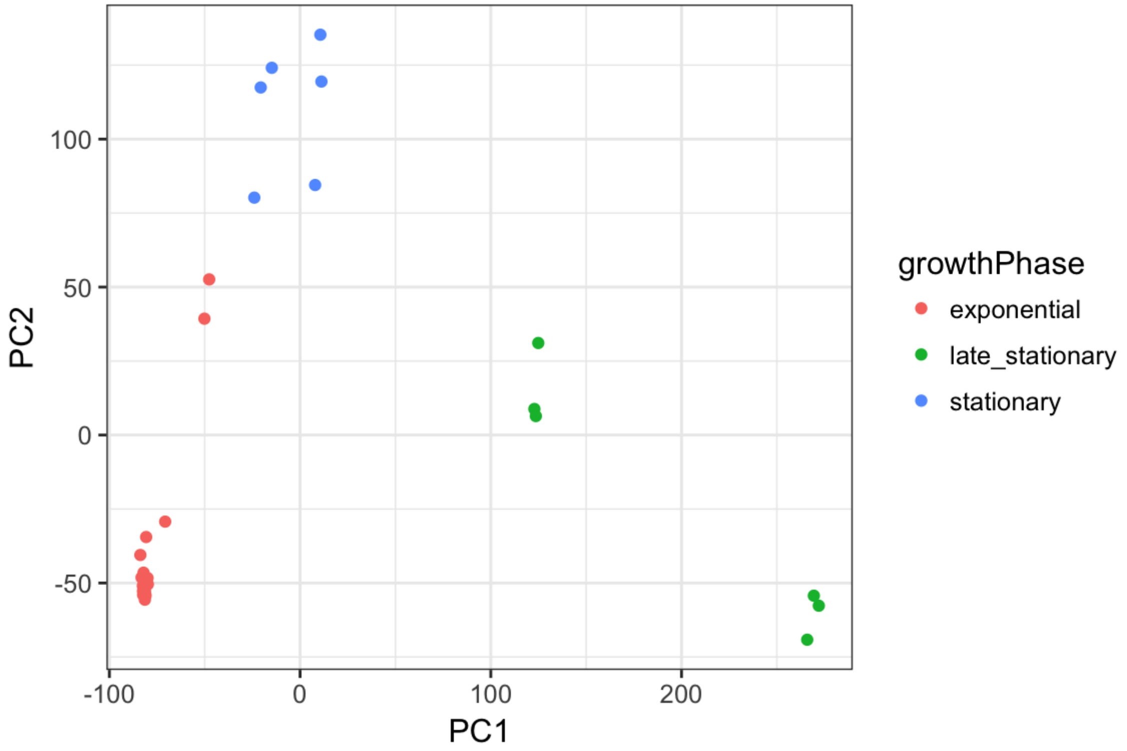
For this homework you will use the `mrna` data set. The `mrna` data set contains gene expression of E.coli grown on minimal medium for two weeks. There are three replicates for each time course. Each row represents observations for different samples. The first five columns correspond to sample name (`dataSet`), carbon source (`carbonSource`), growth time in hours (`growthTime_hr`), bacterial growth phase (`growthPhase`), and batch number (`batchNumber`). The batch number indicates samples that were grown as part of the same replicate. The rest of the columns correspond to gene expression for each gene. The gene names start with ECB (i.e. `ECB_00001`).

```
mrna <- read.csv("http://wilkelab.org/classes/SDS348/data_sets/AG3C_mrna_counts.csv")
mrna <- as_data_frame(mrna) #convert to tibble for easier visualization
mrna
```

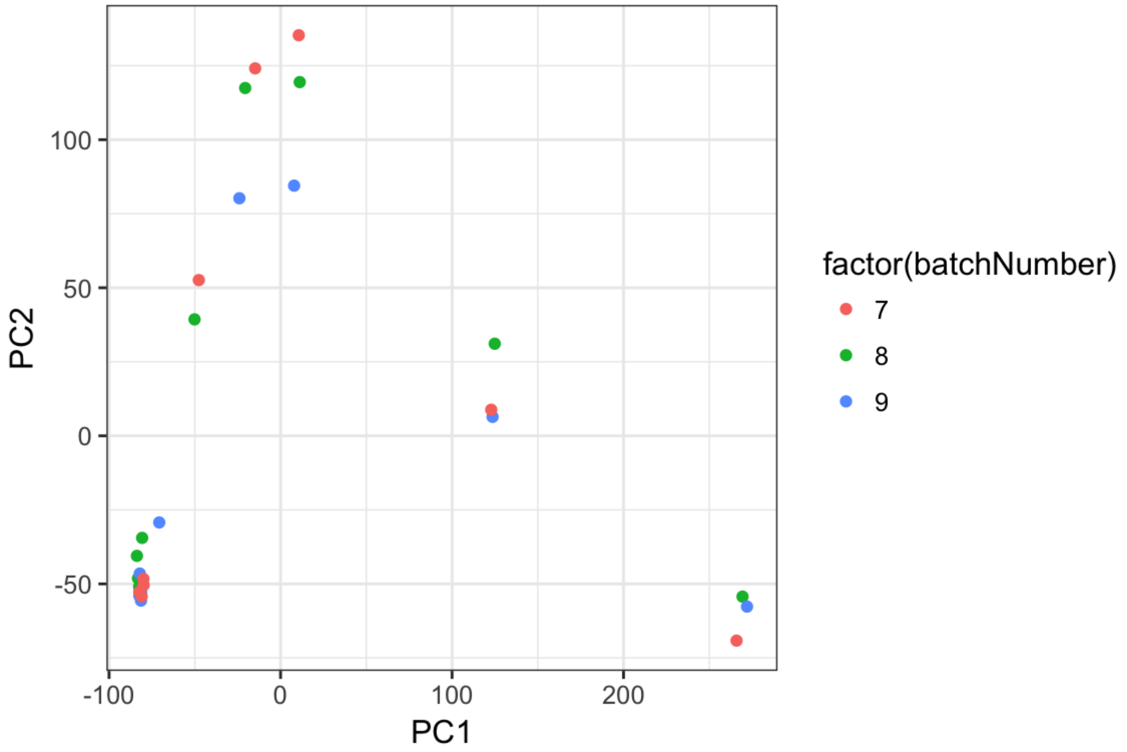
```
## # A tibble: 27 × 4,201
##     dataSet carbonSource growthTime_hr     growthPhase batchNumber
##      <fctr>       <fctr>         <dbl>          <fctr>       <int>
## 1  MURI_016      glucose             3     exponential           8
## 2  MURI_017      glucose             4     exponential           8
## 3  MURI_018      glucose             5     exponential           8
## 4  MURI_019      glucose             6     exponential           8
## 5  MURI_020      glucose             8     exponential           8
## 6  MURI_021      glucose            24      stationary           8
## 7  MURI_022      glucose            48      stationary           8
## 8  MURI_023      glucose           168 late_stationary           8
## 9  MURI_024      glucose           336 late_stationary           8
## 10 MURI_025      glucose             3     exponential           9
## # ... with 17 more rows, and 4196 more variables: ECB_00001 <dbl>,
## #   ECB_00002 <dbl>, ECB_00003 <dbl>, ECB_00004 <dbl>, ECB_00005 <dbl>,
## #   ECB_00006 <dbl>, ECB_00007 <dbl>, ECB_00008 <dbl>, ECB_00009 <dbl>,
## #   ECB_00010 <dbl>, ECB_00013 <dbl>, ECB_00014 <dbl>, ECB_00015 <dbl>,
## #   ECB_00016 <dbl>, ECB_00017 <dbl>, ECB_00018 <dbl>, ECB_00019 <dbl>,
## #   ECB_00020 <dbl>, ECB_00021 <dbl>, ECB_00022 <dbl>, ECB_00023 <dbl>,
## #   ECB_00024 <dbl>, ECB_00025 <dbl>, ECB_00026 <dbl>, ECB_00027 <dbl>,
## #   ECB_00028 <dbl>, ECB_00029 <dbl>, ECB_00030 <dbl>, ECB_00031 <dbl>,
## #   ECB_00032 <dbl>, ECB_00033 <dbl>, ECB_00034 <dbl>, ECB_00035 <dbl>,
## #   ECB_00036 <dbl>, ECB_00037 <dbl>, ECB_00038 <dbl>, ECB_00039 <dbl>,
## #   ECB_00040 <dbl>, ECB_00041 <dbl>, ECB_00043 <dbl>, ECB_00044 <dbl>,
## #   ECB_00045 <dbl>, ECB_00046 <dbl>, ECB_00047 <dbl>, ECB_00048 <dbl>,
## #   ECB_00049 <dbl>, ECB_00050 <dbl>, ECB_00051 <dbl>, ECB_00052 <dbl>,
## #   ECB_00053 <dbl>, ECB_00054 <dbl>, ECB_00055 <dbl>, ECB_00056 <dbl>,
## #   ECB_00057 <dbl>, ECB_00058 <dbl>, ECB_00059 <dbl>, ECB_00060 <dbl>,
## #   ECB_00061 <dbl>, ECB_00062 <dbl>, ECB_00063 <dbl>, ECB_00064 <dbl>,
## #   ECB_00065 <dbl>, ECB_00066 <dbl>, ECB_00067 <dbl>, ECB_00068 <dbl>,
## #   ECB_00069 <dbl>, ECB_00070 <dbl>, ECB_00071 <dbl>, ECB_00072 <dbl>,
## #   ECB_00073 <dbl>, ECB_00074 <dbl>, ECB_00075 <dbl>, ECB_00076 <dbl>,
## #   ECB_00077 <dbl>, ECB_00078 <dbl>, ECB_00079 <dbl>, ECB_00080 <dbl>,
## #   ECB_00081 <dbl>, ECB_00082 <dbl>, ECB_00083 <dbl>, ECB_00084 <dbl>,
## #   ECB_00085 <dbl>, ECB_00086 <dbl>, ECB_00087 <dbl>, ECB_00088 <dbl>,
## #   ECB_00089 <dbl>, ECB_00090 <dbl>, ECB_00091 <dbl>, ECB_00092 <dbl>,
## #   ECB_00093 <dbl>, ECB_00094 <dbl>, ECB_00095 <dbl>, ECB_00096 <dbl>,
## #   ECB_00097 <dbl>, ECB_00098 <dbl>, ECB_00099 <dbl>, ECB_00100 <dbl>,
## #   ECB_00101 <dbl>, ECB_00102 <dbl>, ECB_00103 <dbl>, ...
```

Problem 1 (3 pts): *Perform a principal components analysis (PCA) on gene expression. You do not need to **scale** the data before running PCA because gene expression in this data set have already been normalized. Create a scatterplot of PC1 vs. PC2. First, color each point by bacterial growth phase, and then color each point by batch number. What do you observe? Visually, and without doing any calculations, do the growth phases cluster together in principal-component space? Do the batch numbers cluster together?*

```
mrna %>% select(-dataSet,-growthPhase,-carbonSource, growthTime_hr, batchNumber ) %>% pr
comp() -> pca
mrna.pca <- data.frame(pca$x, growthPhase= mrna$growthPhase, batchNumber= mrna$batchNumb
er)
ggplot(mrna.pca, aes(x=PC1, y=PC2, color=growthPhase)) + geom_point()
```



```
ggplot(mrna.pca, aes(x=PC1, y=PC2, color= factor(batchNumber))) + geom_point()
```

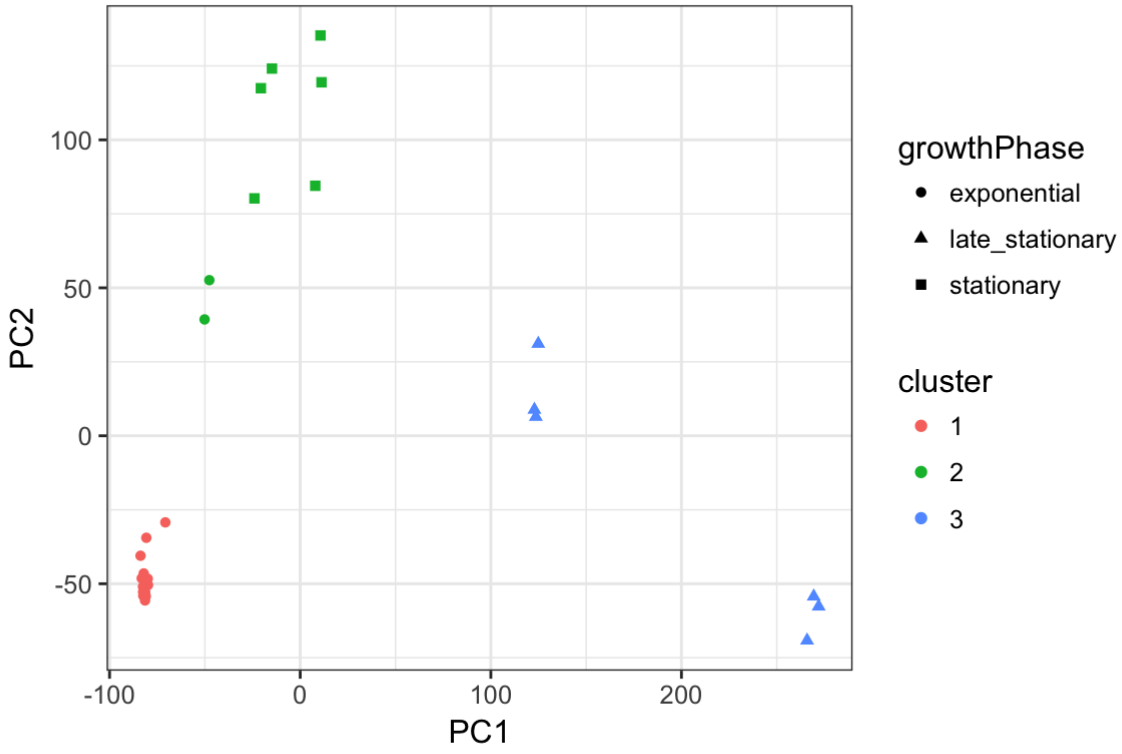


In the growth phase data set I see that the data is clustered seperately based on the stage of growth phase with stationary growth phase having the highest PC2 value, and late stationary growth phase having the largest PC1 value. Visually, in the batch number data set the samples do not cluster based on batch number.

Problem 2 (4 pts): *Now take your matrix of* ***principal components coordinates*** *(not the raw gene expression values!) from Question 1 above and cluster the gene expression into 3 groups (* `centers=3` *) using k-means clustering with 10 random starts (* `nstart=10` *). Create a scatterplot of PC1 vs. PC2. This time, color each point by*

***cluster** and set the plotting symbol by **growth phase**. What do you observe?*

```
pca$x %>% kmeans(centers=3,nstart=10)-> km
mrna_clustered <- data.frame(pca$x, cluster=factor(km$cluster), growthPhase=mrna$growthP
hase)
ggplot(mrna_clustered, aes(x=PC1, y=PC2, color=cluster, shape=growthPhase)) +
geom_point()
```



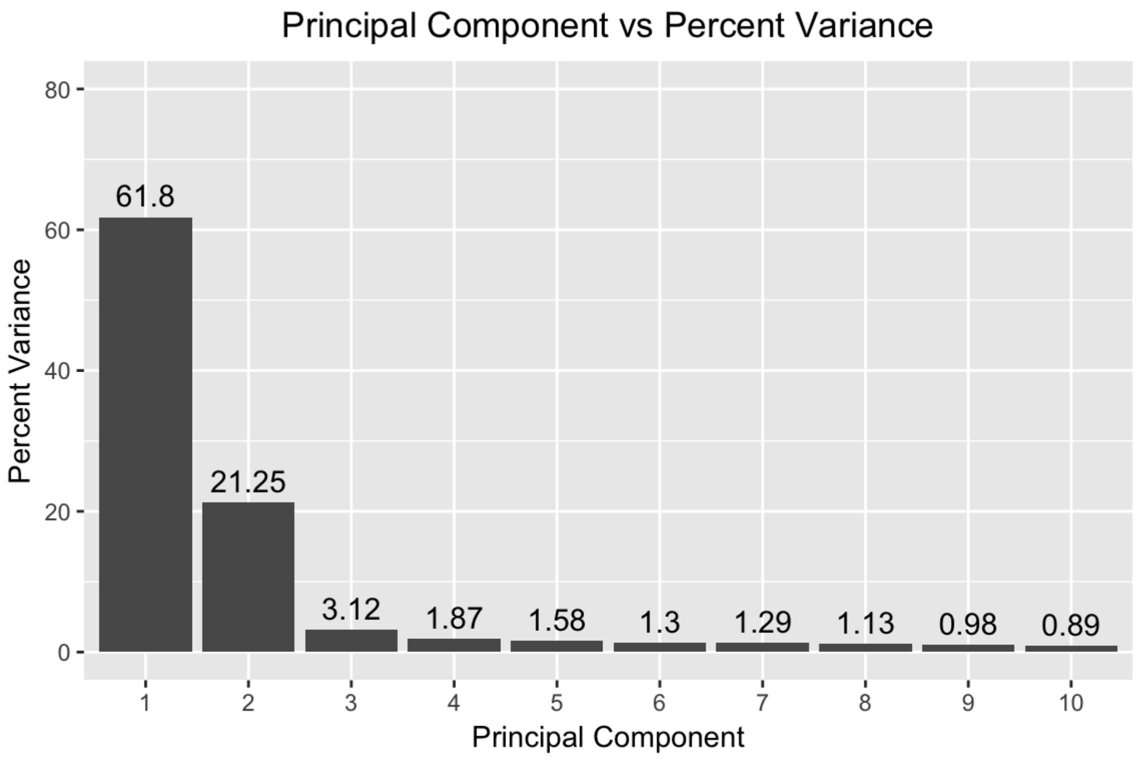
I noticed that most of the data is clustered seperately based on the color and that each data within a cluster has its own shape that represents its growth phase stage.There are a few point in the late stationary growth phase that are clustered in the center of the graph.

Problem 3 (3 pts): *Create a bar plot that shows the percent variance explained by the first 10 principal components. State how much variance is explained by each of the principal components 1 through 4.*

```
percent <- 100*pca$sdev^2/sum(pca$sdev^2)
percent
```

```
##  [1] 6.179780e+01 2.125371e+01 3.123235e+00 1.866917e+00 1.575362e+00
##  [6] 1.295572e+00 1.286193e+00 1.126746e+00 9.833707e-01 8.865781e-01
## [11] 7.990311e-01 7.460609e-01 5.233508e-01 4.207703e-01 3.624319e-01
## [16] 2.989697e-01 2.822579e-01 2.481937e-01 2.175724e-01 1.815069e-01
## [21] 1.582401e-01 1.376409e-01 1.238210e-01 1.165075e-01 9.887934e-02
## [26] 8.928562e-02 1.218747e-29
```

```
perc_data <- top_n(data.frame(percent=percent, PC=1:length(percent)), 10, percent)
ggplot(perc_data, aes(x=factor(PC), y=percent)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=round(percent, 2)), size=4, vjust=-.5) +
  ylim(0, 80) +
  xlab("Principal Component") + ylab("Percent Variance") + ggtitle("Principal Component
 vs Percent Variance") + theme_gray() + theme(plot.title = element_text(hjust = 0.5))
```



In the bar chart, 61.8% variance is explained in component 1, 21.25% variance is explained by component 2, 3.12% variance is explained in component 3, 1.87% variance is explained by component 4.

```
```