StatMethodsHW3A

Joshua Burkhart

January 26, 2016

BMI 651: HW3 A

(1) draft analysis plan for developing classifier for predating developmental stage (E3.25) based on gene expression

Let us assume our final goal is to build a classifier that can be generalized for this microarray platform/probeset to not only correctly label test data as "E3.25" or not in this experiment but whose results may be interpreted by experts in the field to gain knowledge about any biological interactions that occur between selected features and developmental stage. Considering this, we'll limit our feature space to probe intensities, gene expression levels, cell type, cell position label, and genotype. Though we're ignoring EDA and potential batch effects for the moment, we'll still reject metadata such as microarray date, number of cells, or index in the original dataset as feature candidates due to our noted constraint of creating a model that may increase biological insight into development.

1. Extract feature candidates, probe intensities, gene expression levels, etc., from their original data sources and combine them into a single data table. This will allow for simpler handling for the remainder of the analysis.

2. Scan & address missing data. Because we have so few samples (101) and so many total features (over 45,000), we'll prefer to drop features than drop samples. We must scan probe intensities, gene expression levels, etc..

3. Convert genotype to a numeric value. Currently, genotype is stored as one of two values: "FGF4-KO" and "WT". We'll change "FGF4-KO" to 1 and "WT" to 0.

4. Extract development stage and convert it to a numeric value. Currently, Embryonic.day is stored as one of three values: "E3.25", "E3.5", and "E4.5". We'll copy this column to a separate vector and store our positive class, "E3.25", as 1 and our negative class, "E3.5" and "E4.5", as 0.

5. Randomly split data into training and test sets. We should find a way to assure that similar proportians of genotype and class make it into the training and test sets. Also, seeing as we have so few samples, we'll use an 80/20 overall split.

6. Z score transform all the feature values, storing the means and standard deviations for all columns so test data can be transformed similarly later on. Putting features on the same scale allows some optimization algorithms, such as gradient descent, to converge more quickly and generally allows for more direct comparisons between feature distributions.

7. We'd like to explain a lot of variance without using too many features, as that can make things confusing. Normally I'd favor PCA as a logical next step to a learning system but, keeping in mind we'd like to create a model that's interpretable to biologists later on, it may be better to avoid it. Instead, we'll first try selecting features using mutual information and only resort to PCA if results are poor.

8. We'll consider three classification techniques whose results are fairly readily interpretable: decision tree, logistic regression, and support vector machine.

9. Depending on the model libraries available, we'll attempt to optimize parameters using k-fold cross validation, trying k=10. This is an arbitrary choice of approximately 10% of our original data and may have to be reduced if models have a difficult time converging.

10. Test. First, apply Z score transformations to test data using stored means and standard deviations from training data, run the model and see what pops out.

(2) Write R script for performing EDA on the data set and then perform EDA

Data Source Description

a: raw intensity values from the original CEL files arranged in a matrix layout, where each column represents one hybridization, and rows stand for individual array features
x: RMA normalized dataset in the assayData and annotation in the phenoData
xq: single-cell gene expression levels measured by qPCR
xql: single-cell gene expression measured by qPCR, with cells facing the blastocyst cavity labelled fluorescently

Load Data

```
data("x")
data("xq")
data("xql")

#feature candidates
hw3A.x.genotypes <- x@phenoData@data$genotype
hw3A.x.probe_intensities <- data.frame(assayDataElement(x@assayData,'exprs'))
hw3A.xq.cell_type <- xq@phenoData@data$Cell.type
hw3A.xq.gene_expressions <- data.frame(assayDataElement(xq@assayData,'exprs'))
hw3A.xql.label <- xql@phenoData@data$Label
hw3A.xql.gene_expressions <- data.frame(assayDataElement(xql@assayData,'exprs'))

#classes
hw3A.x.classes <- x@phenoData@data$Embryonic.day
hw3A.xq.classes <- xq@phenoData@data$Embryonic.day
hw3A.xql.classes <- xql@phenoData@data$Embryonic.day

#additional data we'll consider when testing for batch effects
hw3A.x.pheno_dates <- as.Date(x@phenoData@data$ScanDate)
hw3A.x.proto_dates <- as.Date(x@protocolData@data$ScanDate)
hw3A.x.num_cells <- as.numeric(x@phenoData@data$Total.number.of.cells)
```

Scan for missing Data

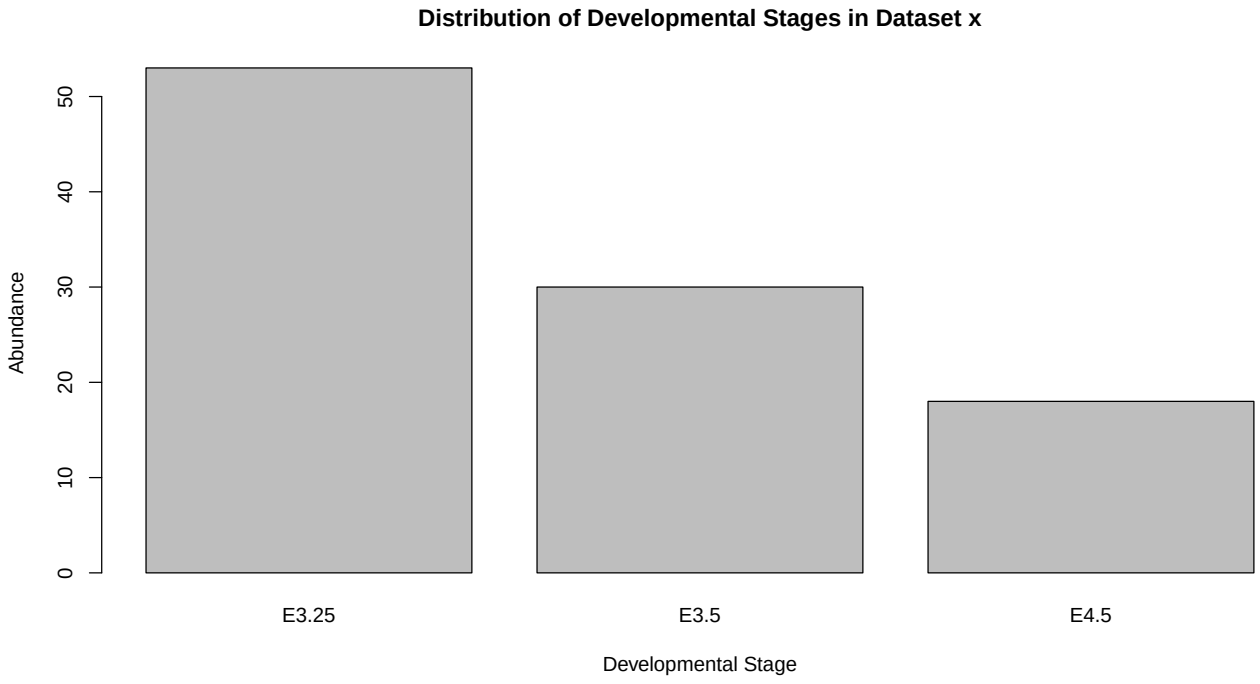
```
sum(is.na(hw3A.x.genotypes))
sum(is.na(hw3A.x.probe_intensities))
sum(is.na(hw3A.xq.cell_type))
sum(is.na(hw3A.xq.gene_expressions))
sum(is.na(hw3A.xql.label))
sum(is.na(hw3A.xql.gene_expressions))

sum(is.na(hw3A.x.classes))
sum(is.na(hw3A.xq.classes))
sum(is.na(hw3A.xql.classes))
```

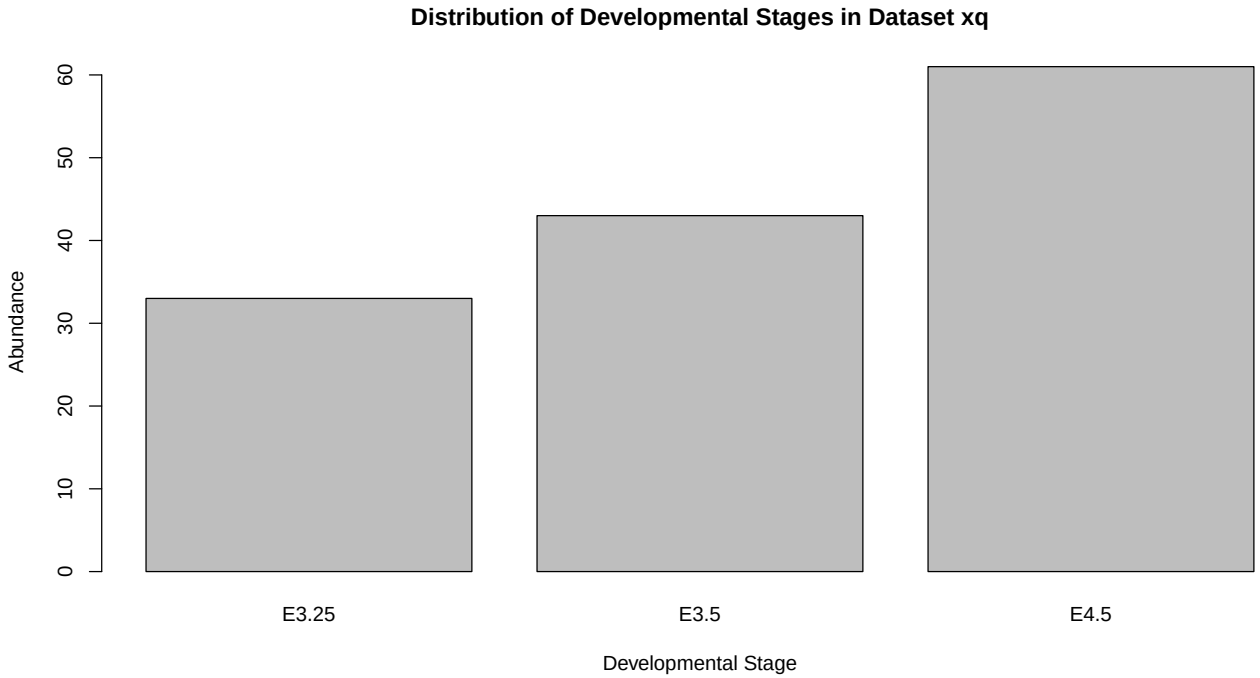
(Output omitted) 302 na's reported in the xq gene expression data but nowhere else. The na's are from Spp1 (40), Prdm14 (40), Sdc4 (40), Morc1 (67), Tbpl1 (2), and Zp3 (113). For now we'll remove them. We may want to replace them with dummy values later as we have so few features for the xq dataset (38 gene expression levels and cell type).

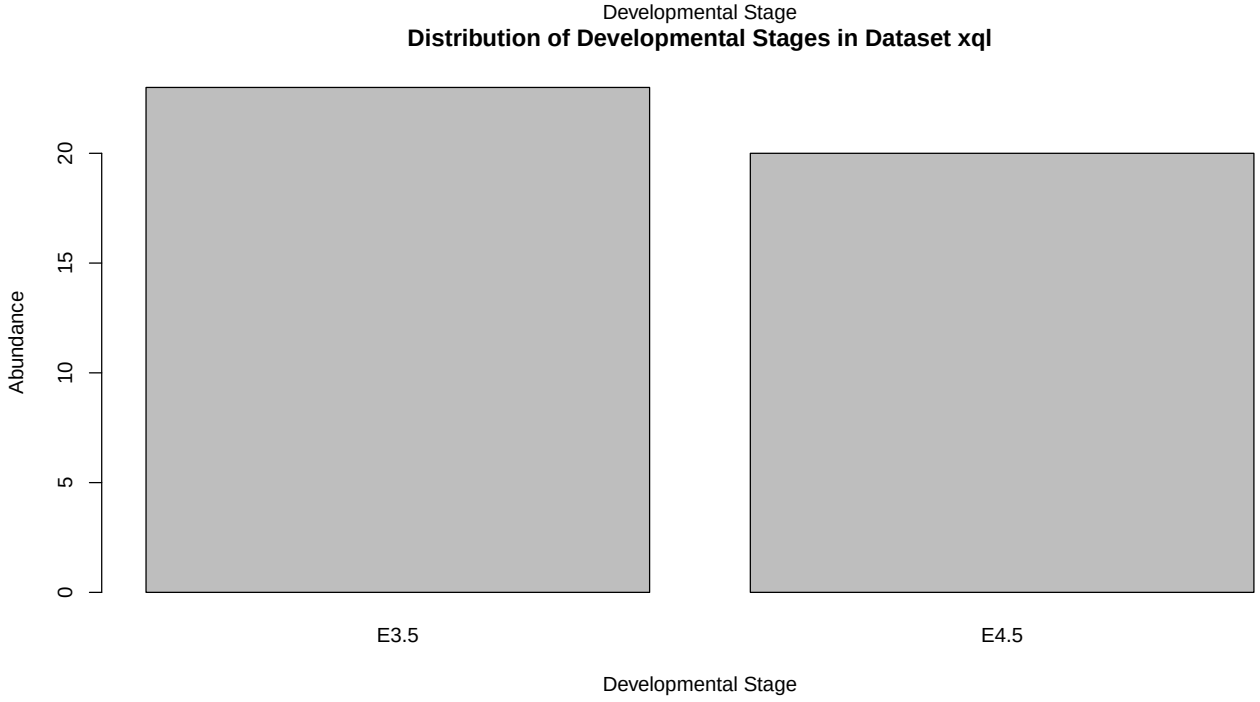
```
t(hw3A.xq.gene_expressions) %>% summary()
hw3A.xq.gene_expressions <- na.omit(hw3A.xq.gene_expressions)
```

Checking Distributions

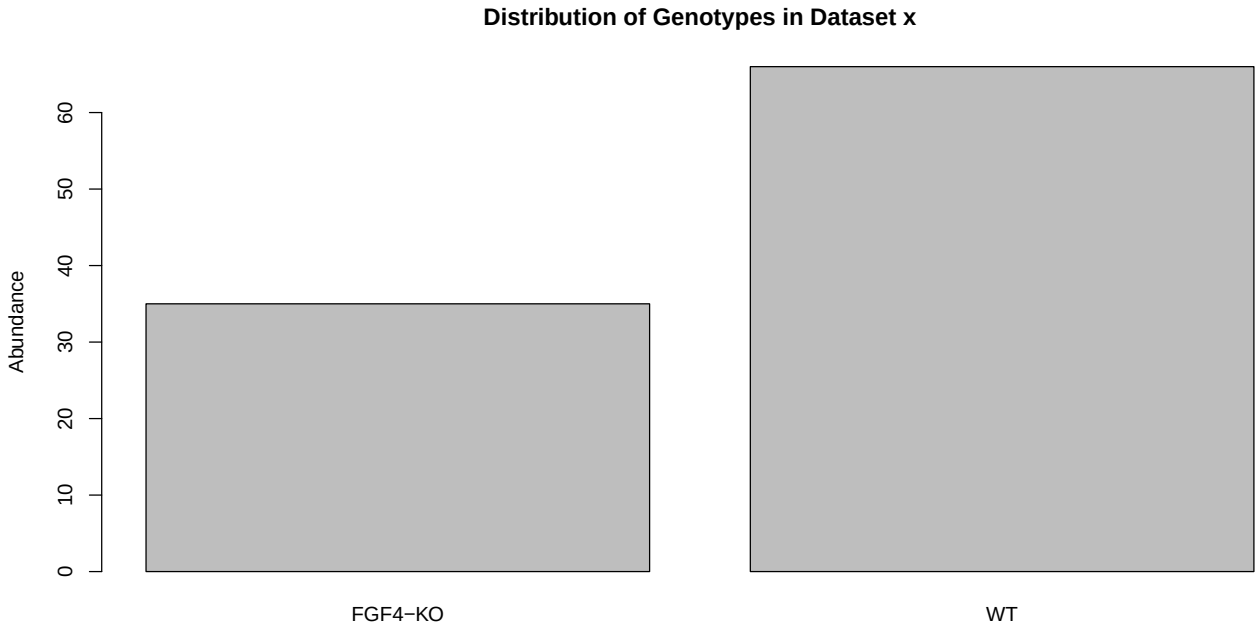
Check outcome ratios. None show a uniform distribution and xql doesn't contain E3.25 developmental stage results. We can safely remove it from our analysis going forward.

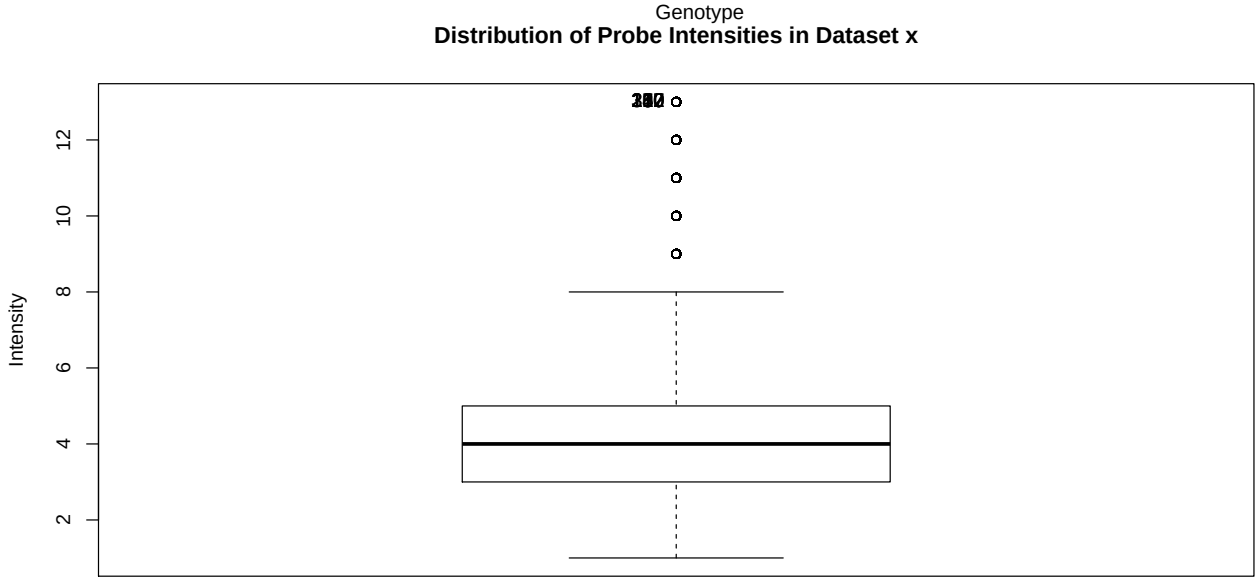


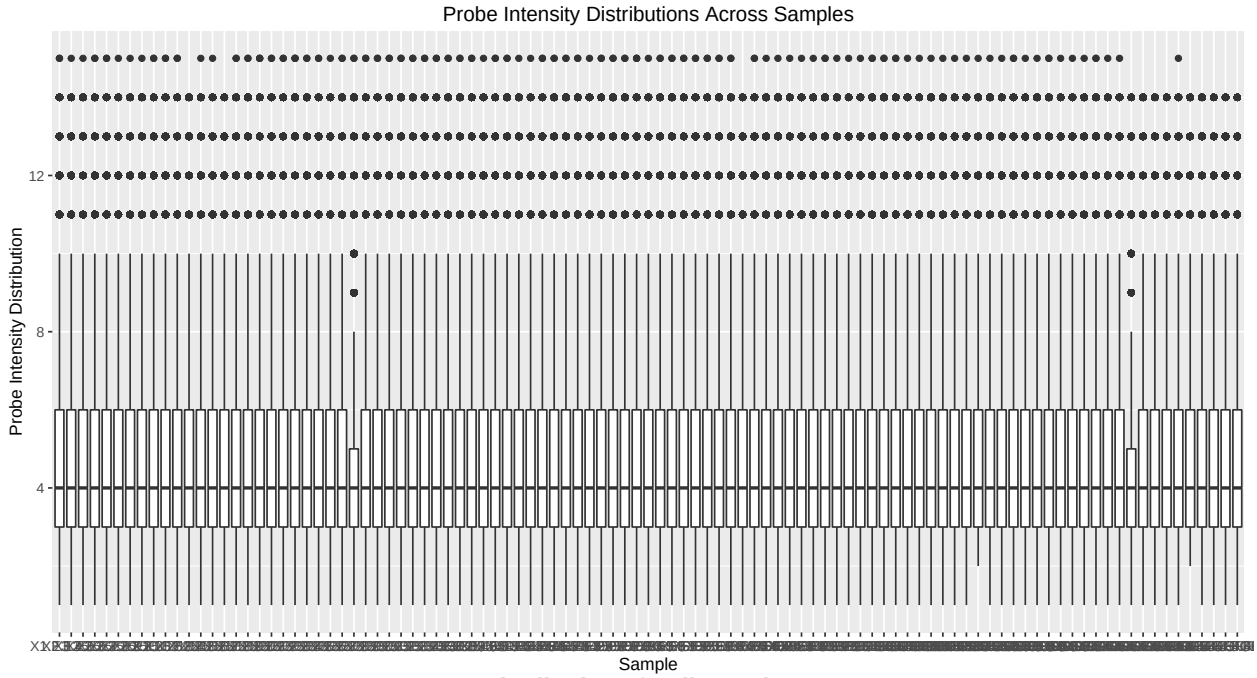


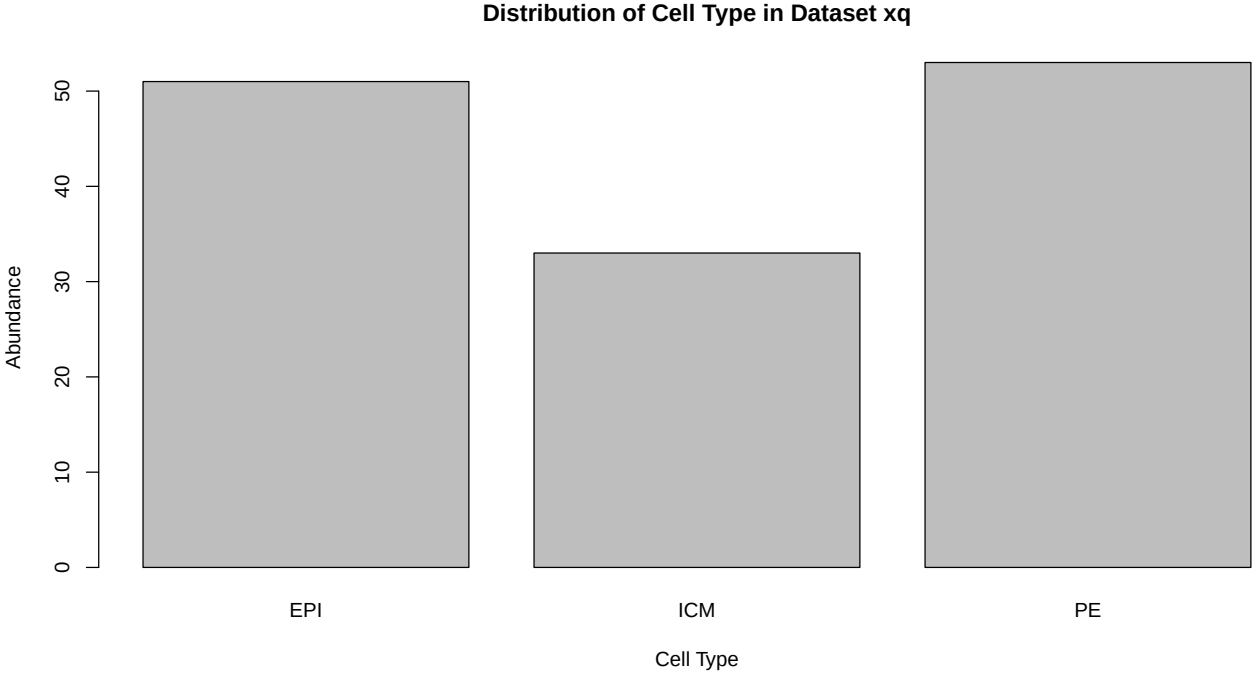


Check feature vectors. Genotypes are skewed. The probe intensities overall are slightly skewed while the probe intensity distributions all appear uniformly distributed with only two having noticable differences. It's odd to have noticably different distributions as they were all supposed to have been normalized in the x dataset. The gene expressions show wide variation among samples.

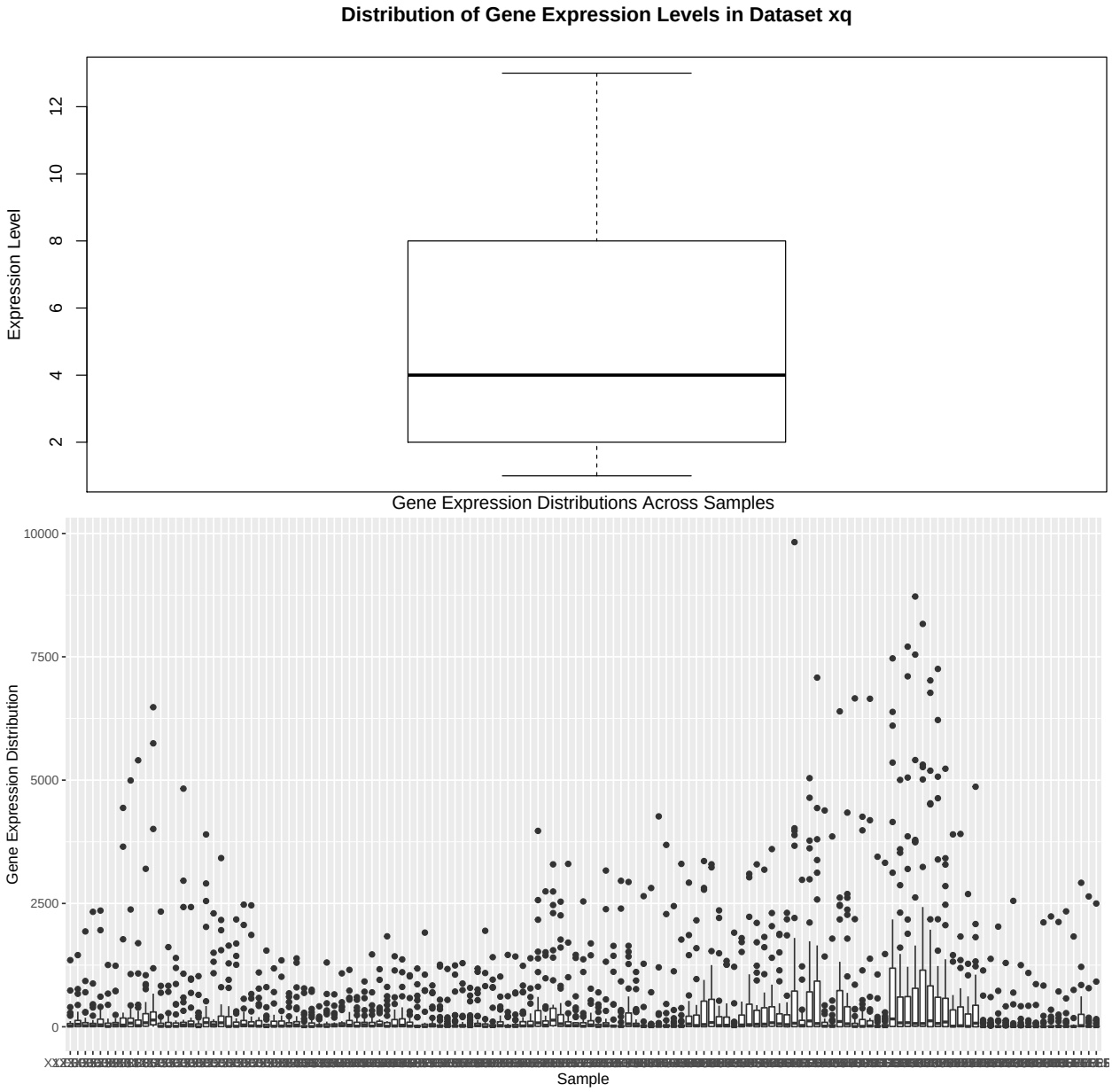

Distribution of Genotypes in Dataset x
Abundance
0
10
20
30
40
50
60
FGF4−KO
WT
Genotype


Distribution of Probe Intensities in Dataset x
Intensity
2
4
6
8
10
12





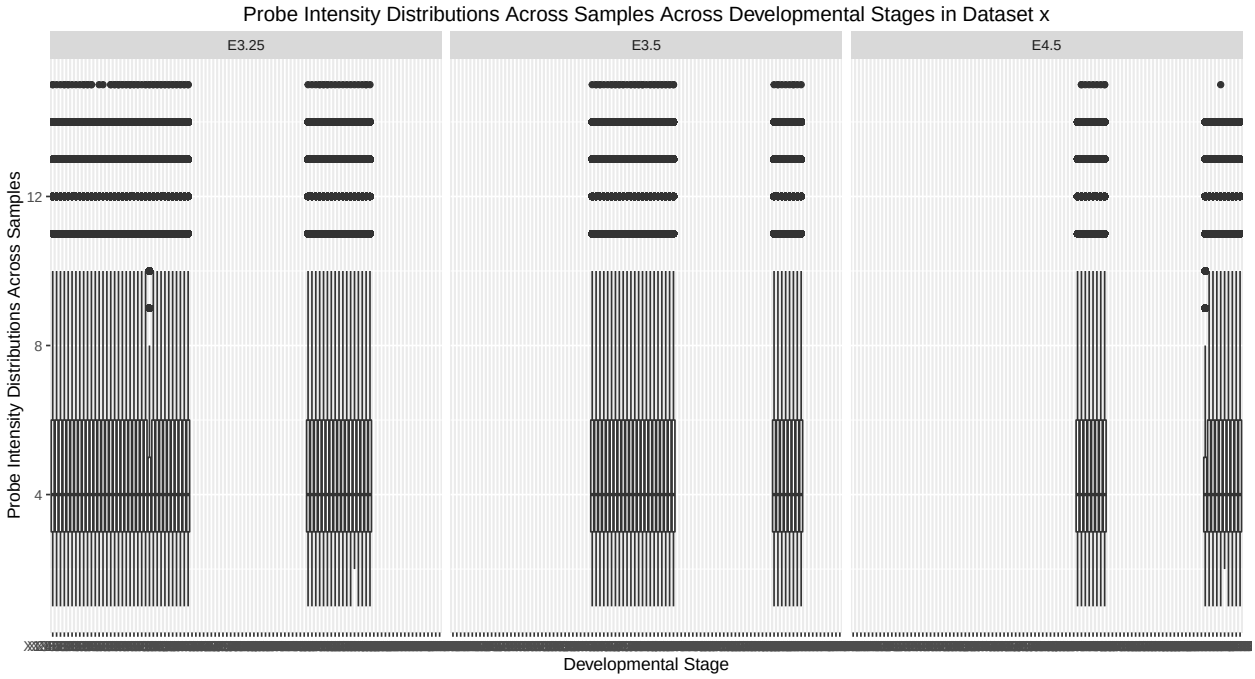

```
Error in 1:rdiff: NA/NaN argument
```



Test distributions vs outcomes. These may not be visualized well but genotypes and cell types can be tested easily using Chi-squared tests. Genotypes are fairly well balanced among developmental stages but cell types are skewed.

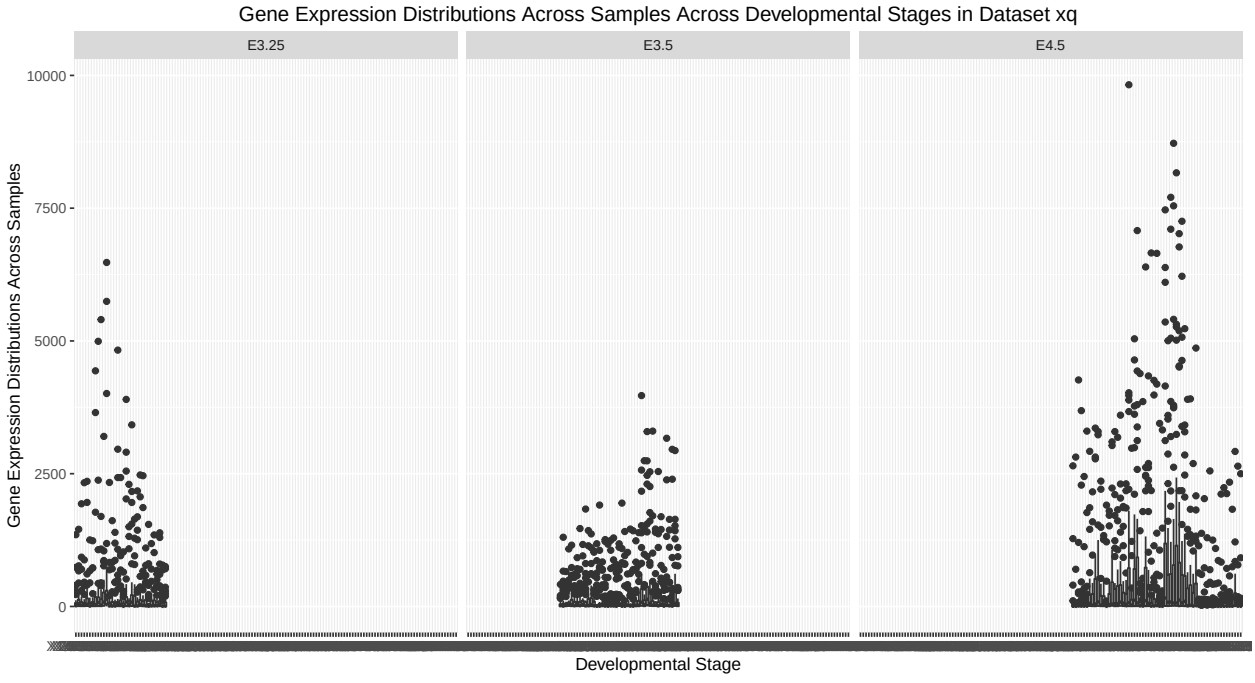
```
#x
table(x=hw3A.x.genotypes,y=hw3A.x.classes) %>% chisq.test()
```

```
    Pearson's Chi-squared test

data:  .
X-squared = 4.4735, df = 2, p-value = 0.1068
```



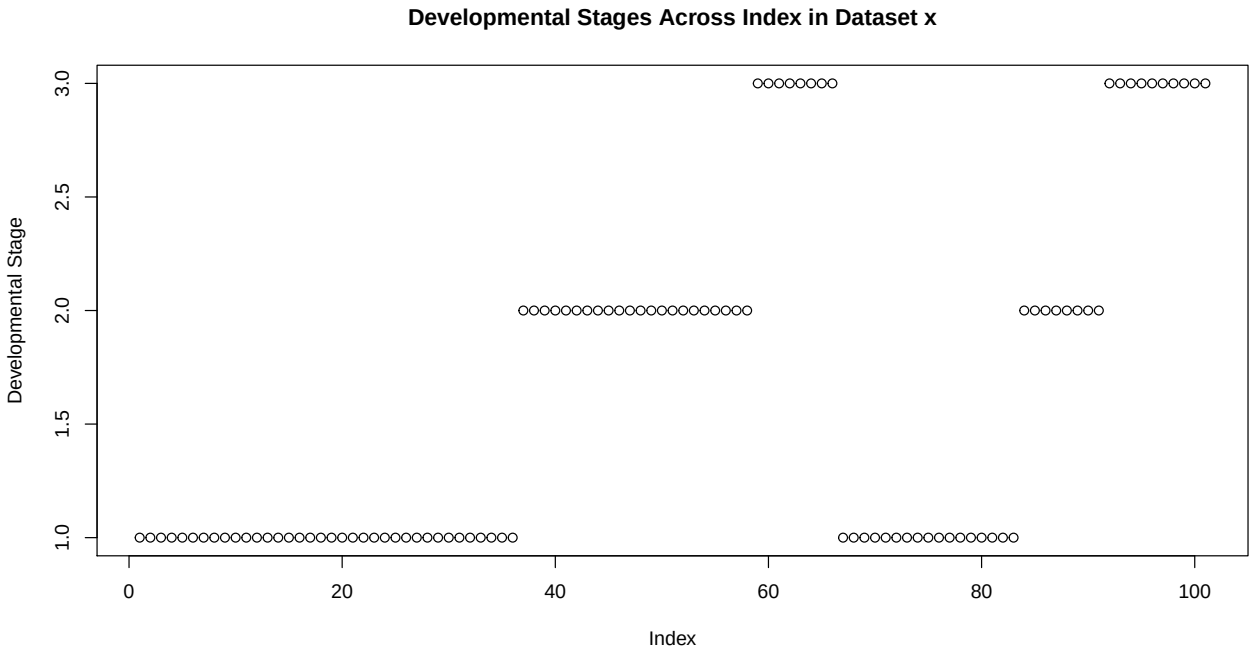

```
#xq
table(hw3A.xq.cell_type,hw3A.xq.classes) %>% chisq.test()
```

```
    Pearson's Chi-squared test

data:  .
X-squared = 137, df = 4, p-value < 2.2e-16
```



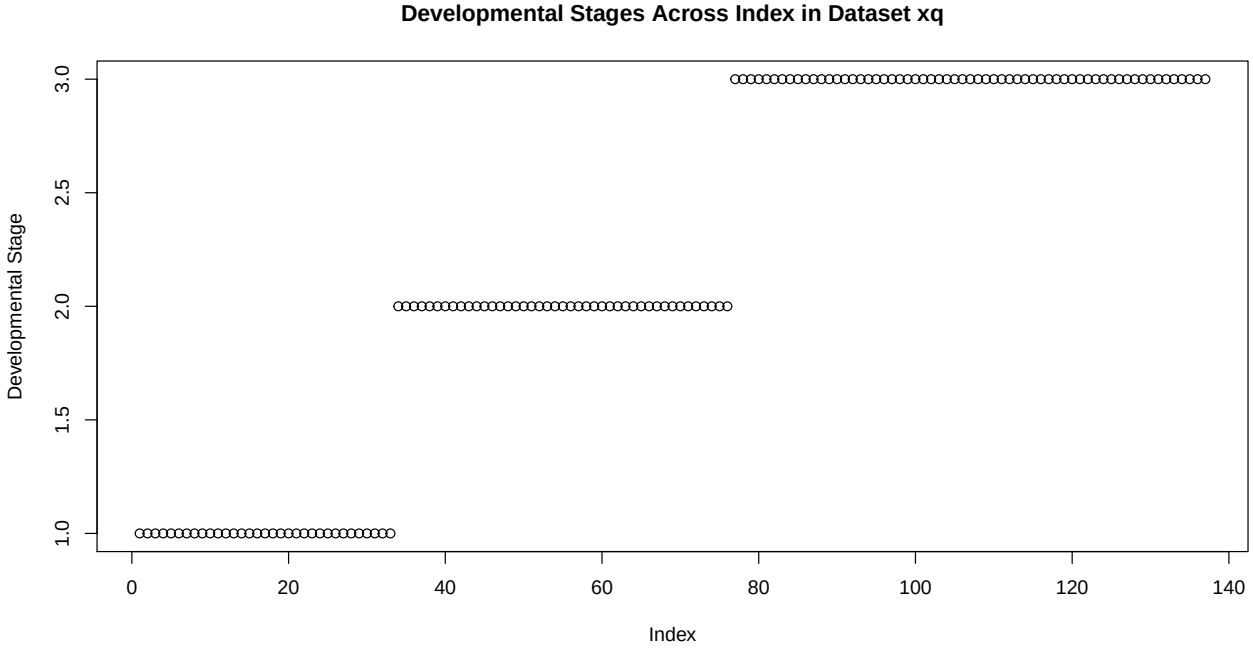
Test for Batch Effects

Plotting array index vs developmental stage. Note xq doesn't store Embryonic.day as a factor like x does. We should consider the regularity of the positions in these datasets when separating our data into training and test sets. Correlation coefficients agree in both of the latter cases, though visual inspection of the first plot also shows obvious bias.




```
cor(x.idx,as.numeric(hw3A.x.classes))
```

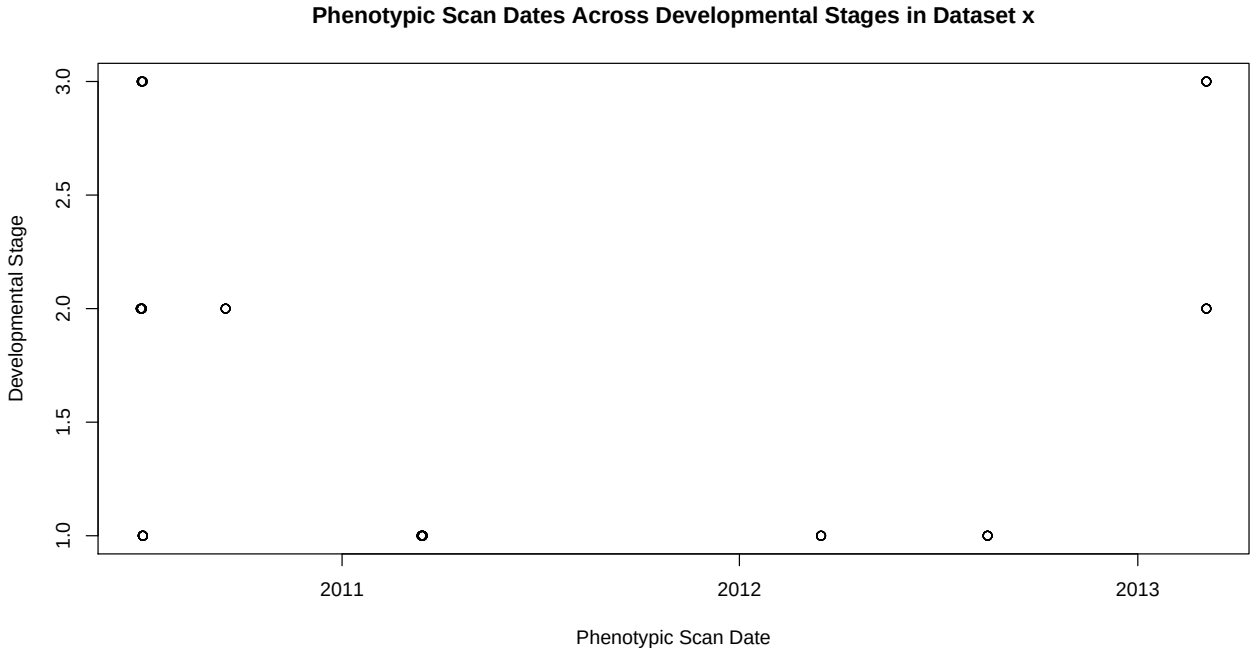
```
[1] 0.582261
```



```
cor(xq.idx,as.numeric(as.factor(hw3A.xq.classes)))
```

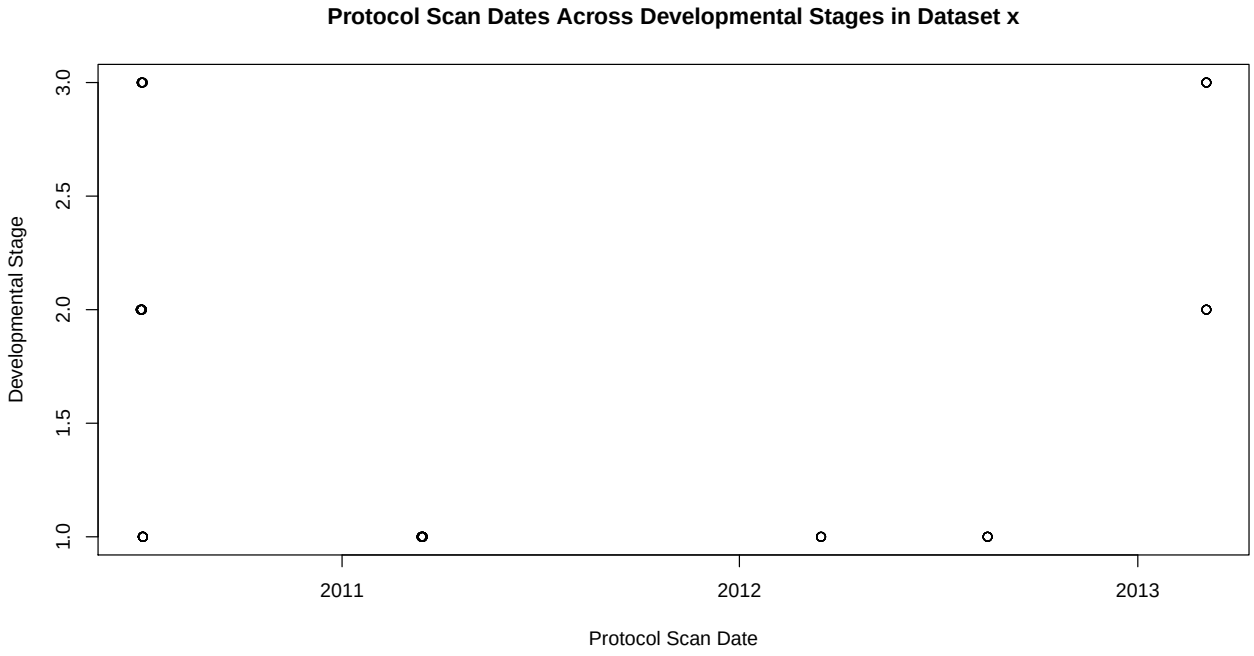
```
[1] 0.9275398
```

Plotting dates vs developmental stage for dataset x. Unfortunately, developmental stage appears highly correlated with both date variables over several years. This indicates unknown and unrecorded variables could be influencing the results of our data. Chi-squared tests agree in both cases, shown by extreemly low p-values.




```
table(hw3A.x.pheno_dates,hw3A.x.classes) %>% chisq.test()
```

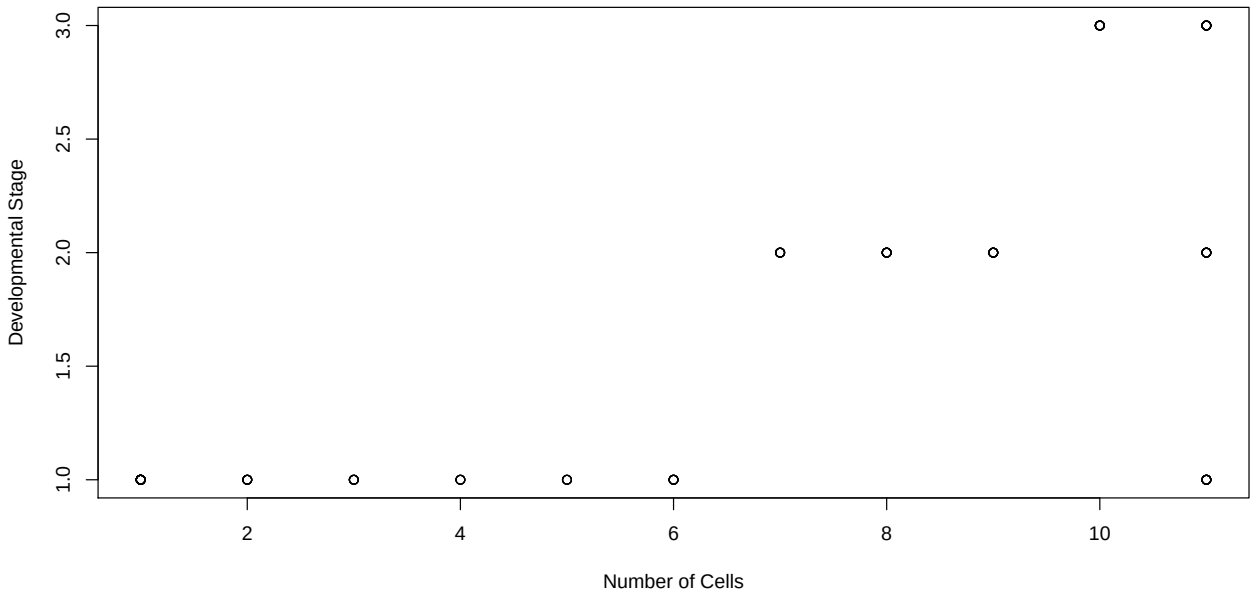
```
    Pearson's Chi-squared test

data:  .
X-squared = 114.83, df = 16, p-value < 2.2e-16
```




```
table(hw3A.x.proto_dates,hw3A.x.classes) %>% chisq.test()
```

```
    Pearson's Chi-squared test

data:  .
X-squared = 96.497, df = 14, p-value = 2.217e-14
```

Plotting number of cells vs developmental stage. Like the dates above, the number of cells appears to correlate well with developmental stage. And again, a Chi-squared test agrees.



```
table(hw3A.x.num_cells,hw3A.x.classes) %>% chisq.test()
```

```
    Pearson's Chi-squared test

data:  .
X-squared = 136.28, df = 20, p-value < 2.2e-16
```

(3) Create version 2 of analysis plan that includes EDA and any revisions (if needed) based on EDA

> Considering the serious batch effects discovered during EDA, we should include a step to address this. Also, it was previously thought that the datasets x, xq, and xql contained the same samples and that their probe intensities, gene expression, etc. could be combined into a single table. After examining the available data, there appears no way to map data from one dataset to another, and xql doesn't have any "E3.25" developmental stage results, thus two separate learning experiments will have to be performed. We'll still use decision tree, logistic regression, and support vector machine for each dataset and still use a training test data split of 80/20.

0. Address batch effects of both scan dates and numbers of cells.
   1-10 (as above)

(4) Write R script for analyzing the data

Transform positive (E3.25) and negative (E3.5 and E4.5) values 1 and 0, respectively

```
hw3A.x.classes <- as.character(hw3A.x.classes)
hw3A.x.classes[hw3A.x.classes == "E3.25"] = 1
hw3A.x.classes[hw3A.x.classes == "E3.5"] = 0
hw3A.x.classes[hw3A.x.classes == "E4.5"] = 0
hw3A.x.classes <- as.factor(hw3A.x.classes)

hw3A.xq.classes <- as.character(hw3A.xq.classes)
hw3A.xq.classes[hw3A.xq.classes == "E3.25"] = 1
hw3A.xq.classes[hw3A.xq.classes == "E3.5"] = 0
hw3A.xq.classes[hw3A.xq.classes == "E4.5"] = 0
hw3A.xq.classes <- as.factor(hw3A.xq.classes)
```

Combine Features for x and xq datasets

```
hw3A.x.features <- rbind(hw3A.x.genotypes,hw3A.x.probe_intensities)
hw3A.xq.features <- rbind(as.factor(hw3A.xq.cell_type),hw3A.xq.gene_expressions)
```

Split into Training & Test sets

```
# x
index <- sample(1:ncol(hw3A.x.features),round(0.8*ncol(hw3A.x.features)))
hw3A.x.features_train <- hw3A.x.features[,index]
hw3A.x.classes_train <- hw3A.x.classes[index]
hw3A.x.features_test <- hw3A.x.features[,-index]
hw3A.x.classes_test <- hw3A.x.classes[-index]

# xq
index <- sample(1:ncol(hw3A.xq.features),round(0.8*ncol(hw3A.xq.features)))
hw3A.xq.features_train <- hw3A.xq.features[,index]
hw3A.xq.classes_train <- hw3A.xq.classes[index]
hw3A.xq.features_test <- hw3A.xq.features[,-index]
hw3A.xq.classes_test <- hw3A.xq.classes[-index]
```

Z-score training set

```
# x
x.train_sd = matrix()
x.train_mean = matrix()
hw3A.x.features_train_normalized <- hw3A.x.features_train
for(i in 2:nrow(hw3A.x.features_train)) # each row is a feature, first is a factor (genotype)
  x.train_sd[i] = sd(hw3A.x.features_train[i,])
  x.train_mean[i] = apply(hw3A.x.features_train[i,],1,mean)
  for(j in 1:ncol(hw3A.x.features_train)) # each column is a sample
    hw3A.x.features_train_normalized[i,j] =
    (hw3A.x.features_train[i,j] - x.train_mean[i]) / x.train_sd[i]

# xq
xq.train_sd = matrix()
xq.train_mean = matrix()
hw3A.xq.features_train_normalized <- hw3A.xq.features_train
for(i in 2:nrow(hw3A.xq.features_train)) # each row is a feature, first is a factor (cell_type)
  xq.train_sd[i] = sd(hw3A.xq.features_train[i,])
  xq.train_mean[i] = apply(hw3A.xq.features_train[i,],1,mean)
  for(j in 1:ncol(hw3A.xq.features_train)) # each column is a sample
    hw3A.xq.features_train_normalized[i,j] =
    (hw3A.xq.features_train[i,j] - xq.train_mean[i]) / xq.train_sd[i]
```

With our datasets now split and Z scored, we are now ready to move on to feature selection (in the next assignment).

References

[1] http://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/
[2] http://stats.stackexchange.com/questions/121522/how-to-transform-test-set-to-the-pca-space-of-the-training-set-i
[3] https://support.bioconductor.org/p/40730/
[4] https://bioconductor.org/packages/release/data/experiment/manuals/Hiiragi2013/man/Hiiragi2013.pdf
[5] http://127.0.0.1:17064/library/Hiiragi2013/doc/Hiiragi2013.pdf
[6] http://www.rdatamining.com/examples/decision-tree