432 Final Project Report - Predicting Cancer Drug Response from Morphological Profile and Protein Level Data

Lam Vo - NetID: lamvo2

12/16/2018

INTRODUCTION AND LITERATURE REVIEW

Motivation

As sequencing technologies make it possible to collect vast amount of data about the cellular environment in complex diseases, the use of statistical learning techniques to integrate and analyze these data sources and guide personalized treatment decisions has attracted great interest from the scientific community. One exciting open challenge is how to accurately predict the potency of drugs on cancer tumors given the tumor cells' omics profiles (such as genomics, transcriptomics, or proteomics).

Different kinds of features representing both the drug and the cancer cells have been used to trained predictive models to address this challenge. While mRNA levels and mutation profiles are popular choices for representing cancer cells, protein levels offer another attractive depiction of the cellular environment 1.

On the other hand, drug molecules have often been represented by structural features (e.g. size and topology) and physical properties (e.g molecular weight, polarity, and lipophilicity). However, the link between these features and the drugs' effects on the cellular environment is rarely clear.

A new emerging method for profiling small molecules' bioactivities is morphological profiling. In this method, a library of compounds are given to a standard cancer cell line, and changes in the cancer cells' morphology (e.g. size and shape of the cell, location of the nuclei and organelles as well as their fluorescent intensities upon staining with fluorescent dyes) after the treatment are measured from microscopic images. The molecules' effects on the cellular environment have been shown to manifest in these changes in morphology, and thus could be used as effective descriptors and predictors for the molecules' bioactivities. Indeed, a number of research groups have been using these features to compare drugs' activity, identify drugs' mechanism of action and side effects, and predicting drugs' performance in biological activity tests 2. However, no studies to date has attempted to use these features for predictive modelling of drug response in cancer.

Analysis Task

My project aims to train a regression model that can take as input a query drug's morphological profile and a query cell line's protein levels (measured by reverse phase protein arrays) and predict the drug's potency on the cancer cell line as measured by area-under-percent-viability-curve (AUC) of that drug-cell line pair. The definition of AUC is illustrated in plot B in Figure 1 3. The horizontal axis is the drug's concentration and the vertical axis is the percent of cells from a population of the cell line killed by the drug. AUC is the red area in the plot. *A high AUC means the drug achieves high killing percentage at a low dose, or in other words, the drug is more potent on the cell line in the drug-cell line pair being investigated.*

As mentioned above, there have been many studies with similar goal, which is to perform regression to predict some metrics of cancer drugs' potency or to classify a cancer cell line as sensitive or insensitive to drugs based on a threshold of these metrics. Besides AUC, another popular metric is IC50, which is the drug concentration at which 50% of the cells are killed (the point on the blue curve in plot A in Figure 1 that correspond to 50% on the vertical axis). A variety of statistical and machine learning algorithms have been explored for this task using data from many publicly available data sets. Some of the most well-known ones include the Cancer Cell Line Encyclopedia CCLE, the Catalogue of Somatic Mutations in Cancer COSMIC, the Genomic of Drug Sensitivity in Cancer database GDSC. Investigated models include variations of regularized linear models trained on COSMIC and CCLE data sets 4, variations of random

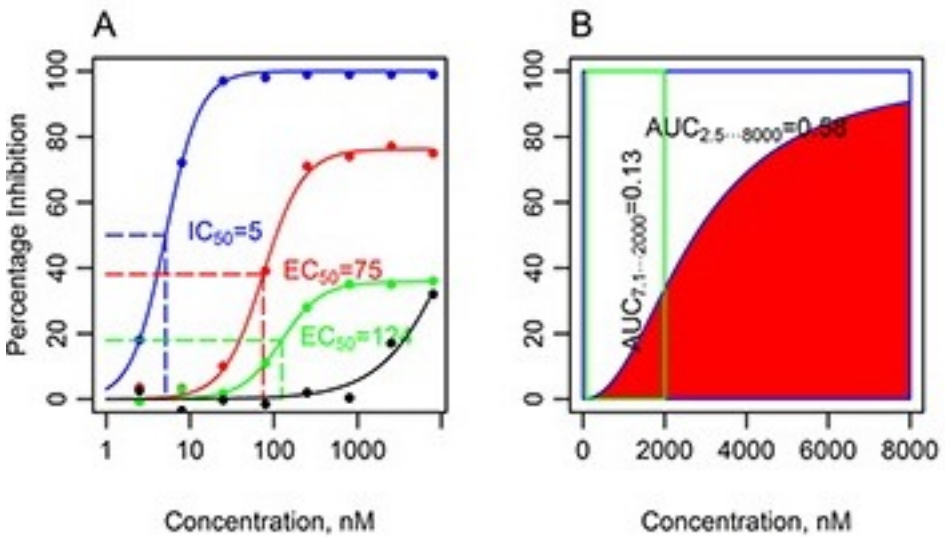


Figure 1: AUC definition

forest models trained on NCI-60 cell line data 5 and CCLE and GDSC data 6, deep neural network trained on GDSC data 7. My data comes from the Cancer Therapeutics Response Portal CTRP, of which I have not been able to find previous drug response prediction studies.

EXPLORING THE DATA SETS

Note: The data sets explored below have gone through cleaning and transformation. I am happy to provide the raw data and Python notebooks used for preprocessing upon request.

Drug Data

The drugs' morphological profiles were extracted from GigaDB at this link and missing and duplicate data points have been removed. For every row, the first column is the drug' master ID in the CTRP database, and the remaining columns contain measurements for morphological properties of U2OS (an osteosarcoma cell line) cells upon treatment with that drug. There are 80 drugs and 1532 morphological properties.

```
drugs = read.csv('drugs.csv')
head(drugs[, 1:3])
```

```
##   Master.ID Cells_AreaShape_Area Cells_AreaShape_Center_X
## 1    417262           0.10068269              -0.01068128
## 2    415688           0.06699153               0.05301241
## 3     27894          -0.22886003               0.02317770
## 4     50715          16.94456642               0.40456857
## 5    414479           0.22325481               0.02705412
## 6     25393           0.26196895               0.03207342
```

```
dim(drugs)
```

```
## [1]   80 1533
```

```
sum(is.na(drugs)) # check for missing data points
```

```
## [1] 0
```

```
sum(duplicated(drugs)) # check for duplicate data points.
```

```
## [1] 0
```

A quick look at some of the morphological features

```
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.4.4
```

```
skim(drugs[, 2:10])
```

```
## Skim summary statistics
##  n obs: 80
##  n variables: 9
```

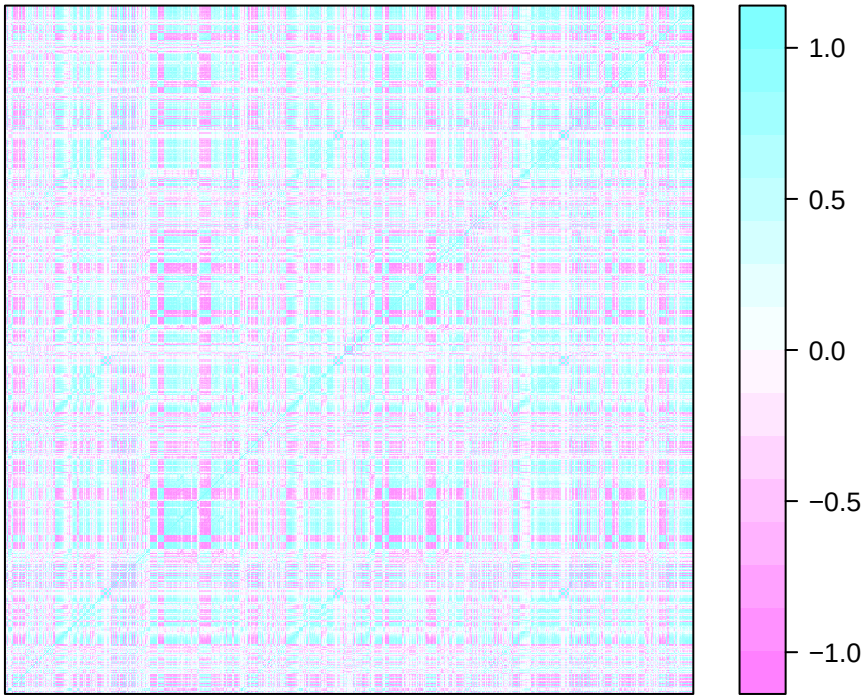
```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
##
## -- Variable type:numeric -------------------------------------------------------------------
##                          variable missing complete  n    mean    sd    p0
##              Cells_AreaShape_Area       0       80 80  0.89   3.32  -1.52
##           Cells_AreaShape_Center_X       0       80 80  0.021  0.075 -0.15
##           Cells_AreaShape_Center_Y       0       80 80 -0.0077 0.053 -0.2
##        Cells_AreaShape_Compactness       0       80 80  0.075  0.37  -1.19
##       Cells_AreaShape_Eccentricity       0       80 80 -0.058  0.54  -2.63
##             Cells_AreaShape_Extent       0       80 80  0.11   0.49  -0.54
##         Cells_AreaShape_FormFactor       0       80 80  0.3    0.75  -1.13
##    Cells_AreaShape_MajorAxisLength       0       80 80  0.28   0.78  -1.8
##   Cells_AreaShape_MaxFeretDiameter       0       80 80  0.25   0.78  -1.92
##      p25     p50   p75   p100     hist
##  -0.032   0.11   0.38  18.42
##  -0.016   0.009  0.035  0.4
##  -0.032  -0.0032 0.025  0.093
##  -0.019   0.11   0.22   1.02
##  -0.0066  0.086  0.16   0.57
##  -0.097  -0.022  0.061  1.92
##  -0.054   0.069  0.36   3.25
##  -0.023   0.15   0.43   3.41
##  -0.033   0.13   0.42   3.41
```

Plot the correlation matrix for drug data.

```
library(lattice)
corMatDrug = cor(drugs[, 2:1533])
dimnames(corMatDrug) = list(rep('', ncol(corMatDrug)), rep('', ncol(corMatDrug)))
```

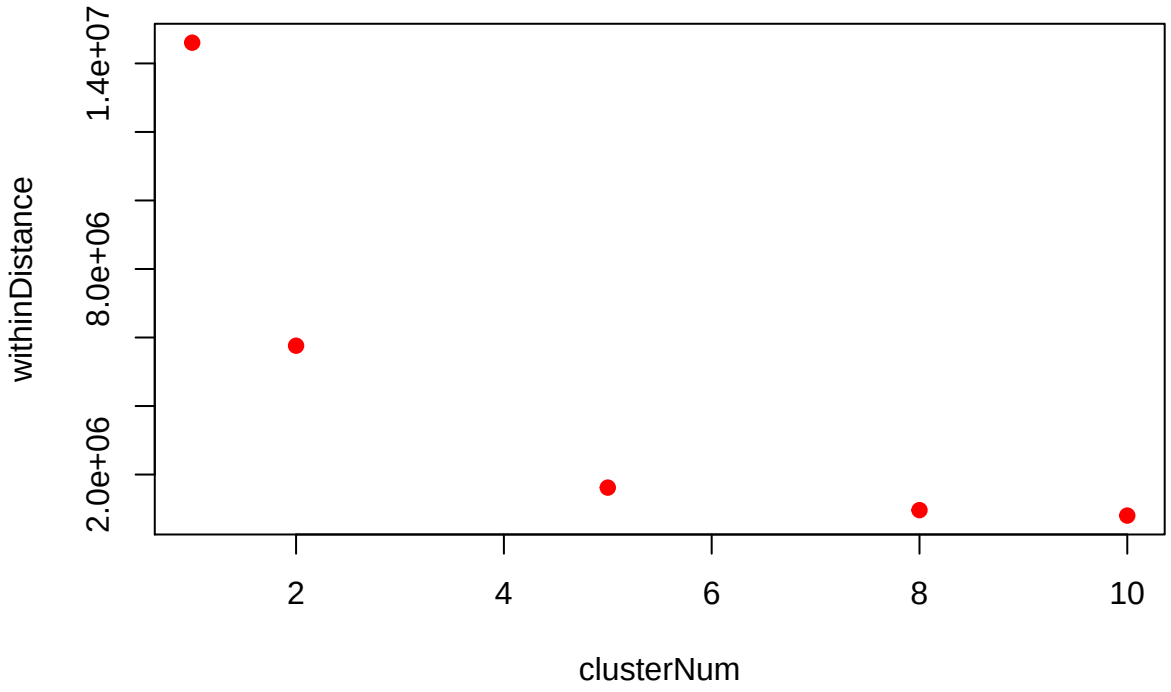
```
levelplot(corMatDrug, xlab = NULL, ylab = NULL, scale = list(tck = c(0, 0)))
```



It is evident from the heatmap that most of the drug features are either correlated or anti-correlated.

Apply k-means clustering to the drug data

```
clusterNum = c(1,2,5,8,10)
withinDistance = rep(1,5)
for (i in c(1:length(clusterNum))){
  drugClusters = kmeans(drugs[, 2:1533], clusterNum[i], nstart =20)
  withinDistance[i] = drugClusters$tot.withinss
}
plot(clusterNum, withinDistance, col = 'red', pch = 19)
```



Based on the elbow plot, choosing to cluster the drugs into 5 clusters is a good decision. Perform k-means clustering for the drugs using k = 5 (k was selected from elbow plot)

```
set.seed(101)
drugClusters = kmeans(drugs[, 2:1533], 5, nstart = 20)
```

Take a look at the sizes of the drug clusters.

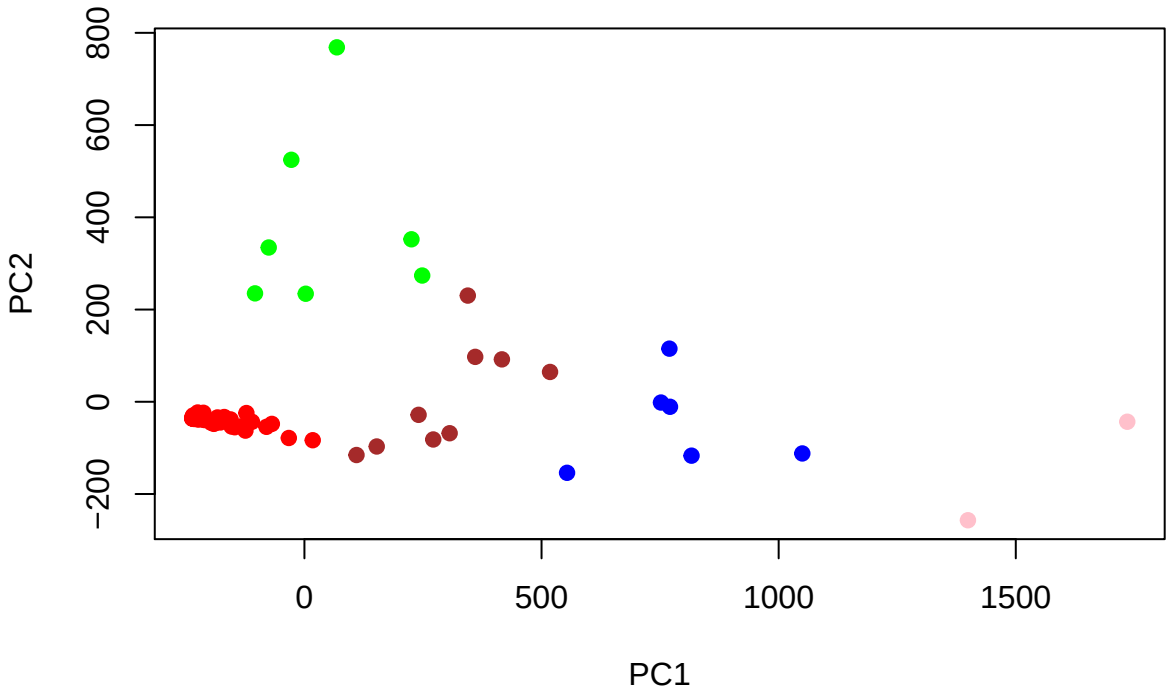
```
drugClusters$size
```

```
## [1] 56  7  6  2  9
```

Visualize the drug clusters with PCA

```
drugPCAs = prcomp(drugs[, 2:1533])
```

```
plot(drugPCAs$x[, 1:2], col = c('red','green','blue','pink','brown')[drugClusters$cluster], pch = 19)
```



This plot reveals clear drug clusters in the drug data.

Cell Line Data The cell lines' protein levels data were downloaded from the Cancer Cell Line Encyclopedia CCLE and removed missing and duplicate data points. For every row, the first column is the name of the cell line and the subsequent columns are the names of the proteins whose expression levels are being measured. There are 899 cell lines and 214 proteins.

```
cellProteins = read.csv('cellProteins.csv')
head(cellProteins[, 1:3])
```

```
##   Cell.Line X14.3.3_beta X14.3.3_epsilon_Caution
## 1     DMS53    -0.104888                0.060414
## 2    SW1116     0.358504               -0.180291
## 3  NCIH1694     0.028738                0.071902
## 4     P3HR1     0.120039               -0.066802
## 5     HUT78    -0.268997               -0.060281
## 6     UMUC3    -0.171170                0.055813
```

```
dim(cellProteins)
```

```
## [1] 899 215
```

```
sum(is.na(cellProteins)) # check for missing data points
```

```
## [1] 0
```

```
sum(duplicated(cellProteins)) # check for duplicate data points.
```

```
## [1] 0
```

A quick look at some of the protein level features.

```
skim(cellProteins[, 2:10])
```

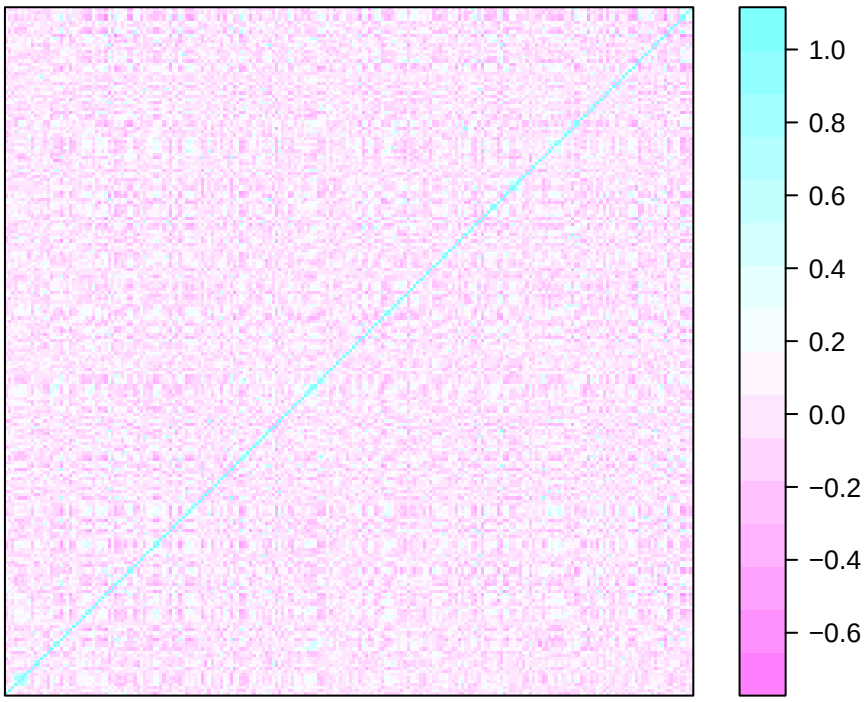
```
## Skim summary statistics
##  n obs: 899
##  n variables: 9
```

```
##
## -- Variable type:numeric -------------------------------------------------------------------
##                  variable missing complete   n    mean   sd    p0   p25 p50
##      A.Raf_pS299_Caution       0      899 899  0.052  0.35 -0.79 -0.18   0
##             X14.3.3_beta       0      899 899  0.021  0.2  -0.54 -0.11   0
##  X14.3.3_epsilon_Caution       0      899 899  0.017  0.2  -0.46 -0.11   0
##             X14.3.3_zeta       0      899 899  0.046  0.43 -1.74 -0.22   0
##                  X4E.BP1       0      899 899  0.016  0.56 -1.63 -0.37   0
##             X4E.BP1_pS65       0      899 899  0.047  0.57 -1.58 -0.35   0
##         X4E.BP1_pT37_T46       0      899 899  0.0075 0.68 -2.04 -0.48   0
##             X4E.BP1_pT70       0      899 899  0.017  0.33 -1.11 -0.18   0
##                   X53BP1       0      899 899 -0.038  0.68 -3.97 -0.41   0
##   p75 p100     hist
##  0.22 1.99
##  0.13 1.2
##  0.12 2.26
##  0.29 1.67
##  0.4  1.82
##  0.4  2.29
##  0.46 2.4
##  0.19 1.56
##  0.38 2
```

Plotting the correlation matrix for cell line data.

```
library(lattice)
corMatCell = cor(cellProteins[, 2:215])
dimnames(corMatCell) = list(rep('', ncol(corMatCell)), rep('', ncol(corMatCell)))
```

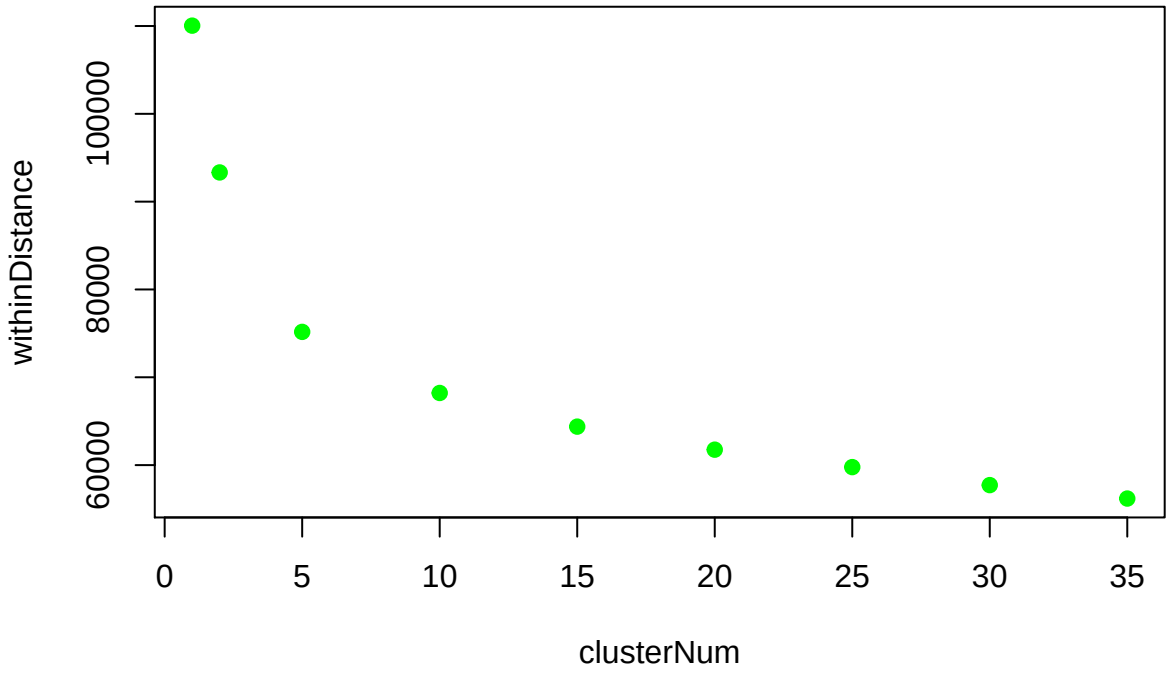
```
levelplot(corMatCell, xlab = NULL, ylab = NULL, scale = list(tck = c(0, 0)))
```



It is evident from the heatmap that most of the protein levels are either non-correlated or anti-correlated.

Apply k-means clustering to the cellLines data

```
clusterNum = c(1,2,5,10,15,20,25,30,35)
withinDistance = rep(1,9)
for (i in c(1:length(clusterNum))){
  cellClusters = kmeans(cellProteins[, 2:215], clusterNum[i], nstart =20)
  withinDistance[i] = cellClusters$tot.withinss
}
plot(clusterNum, withinDistance, col = 'green', pch = 19)
```



Based on the elbow plot, choosing to cluster the cell lines into 10 clusters is a good decision.

Perform k-means clustering for cells (k was selected from elbow plot)

```
set.seed(101)
cellClusters = kmeans(cellProteins[, 2:215], 10, nstart = 20)
```

Take a look at the sizes of cell line clusters.

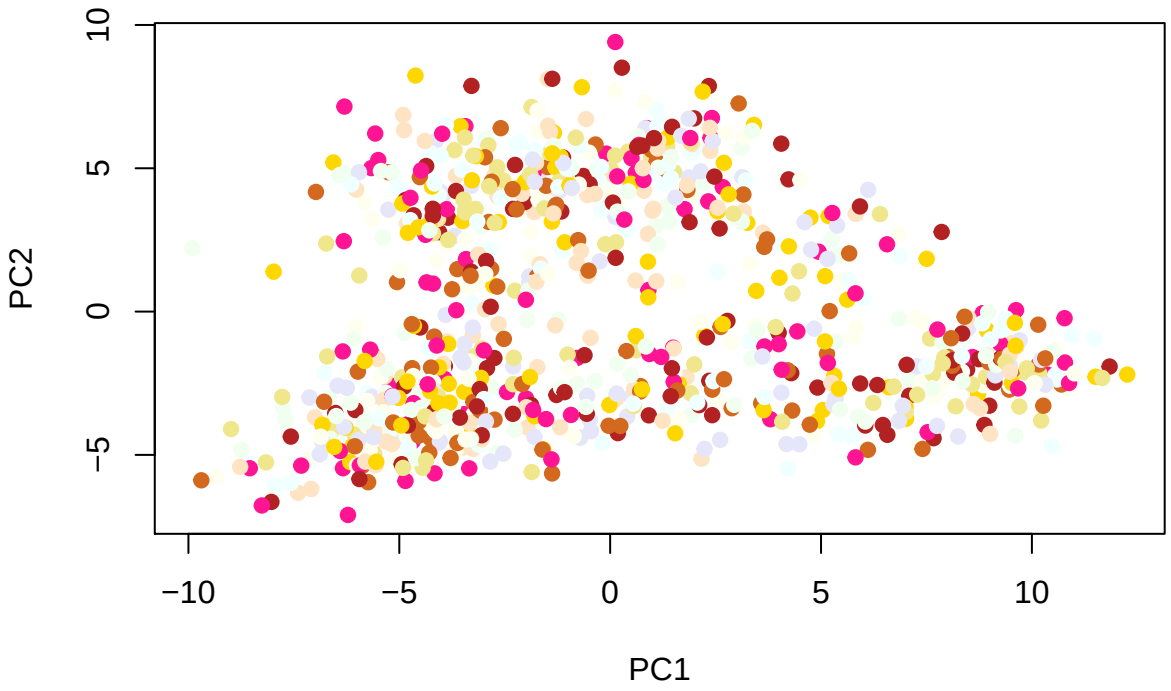
```
cellClusters$size
```

```
##  [1]  18 222  26 103  77  56  69 100 106 122
```

Visualize the cell line clusters using PCA.

```
cellPCAs = prcomp(cellProteins[, 2:215])
```

```
plot(cellPCAs$x[, 1:2], col = c('azure','bisque','chocolate', 'deeppink','firebrick','gold','honeydew',
```



Drug Response Data

The data on drugs' potency on cell lines were downloaded from the Cancer Therapeutics Response Portal CTRP and missing and duplicate data points were removed. For every row, the first column contains AUC value, the second column is the drug's master ID in the CTRP database, and the third column is the name of the cell line treated with the drug. There are 54805 observations.

```
AUCs = read.csv('AUCs.csv')
head(AUCs)
```

```
##      AUC Master.ID Cell.Line
## 1 13.390     23256      CAS1
## 2 14.385     25036      CAS1
## 3 15.570     26870      CAS1
## 4 13.510     26914      CAS1
## 5 13.149     26972      CAS1
## 6 12.430     27894      CAS1
```

```
dim(AUCs)
```

```
## [1] 54810     3
```

```
sum(is.na(AUCs)) # check for missing data points
```

```
## [1] 0
```

```
sum(duplicated(AUCs)) # check for duplicate data points.
```

```
## [1] 5
```

Summary statistics for AUC values

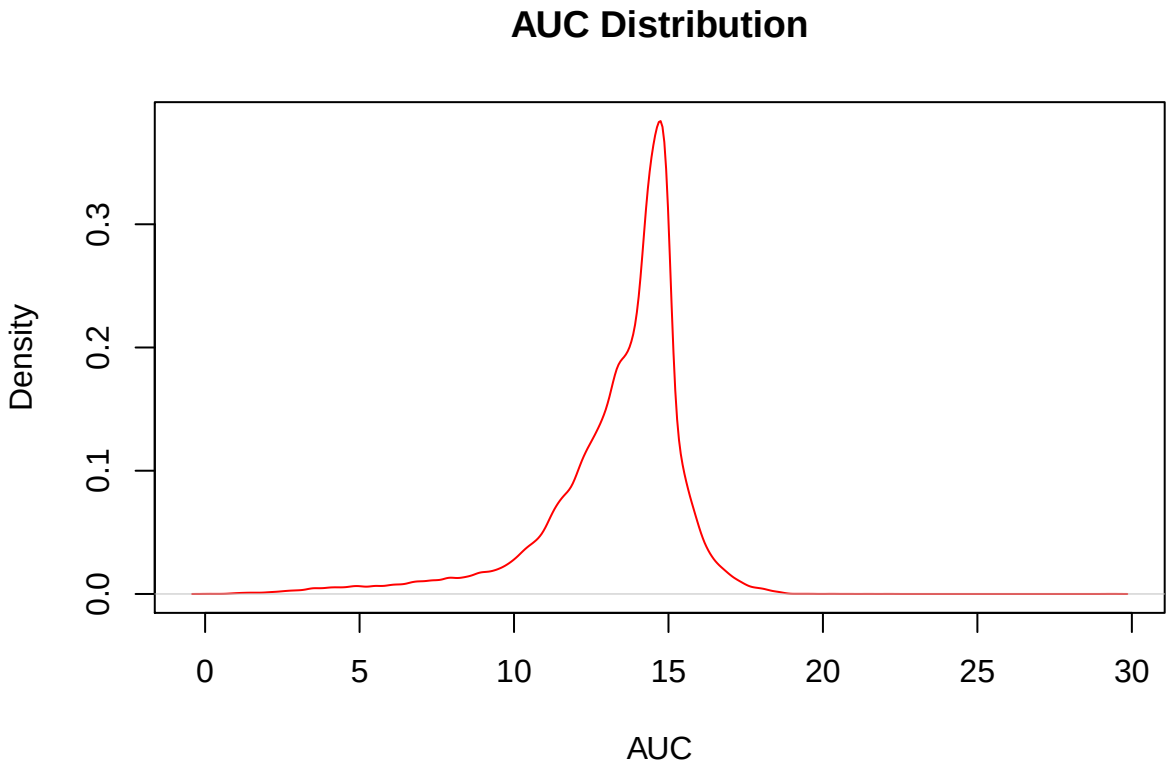
```
skim(AUCs$AUC)
```

```
##
## Skim summary statistics
##
```

```
## -- Variable type:numeric -------------------------------------------------------------------
##  variable missing complete     n  mean   sd    p0   p25 p50   p75  p100
##  AUCs$AUC       0    54810 54810 13.36 2.29 0.084 12.55  14 14.76 29.35
##      hist
##
```

Plotting AUC distribution

```
plot(density(AUCs$AUC), xlab = 'AUC', main = 'AUC Distribution', col = 'red')
```



The AUC values are quite normally distributed.

I have merged these 3 data frames into one big one. After removing duplicated rows, this data frame contains 49536 drug-cell line pairs, the identities of the drug and the cell line in each pair, their corresponding AUC values, and features of both the drugs and the cell lines.

```
big = merge(AUCs, drugs, all = FALSE)
big = merge(big, cellProteins, all = FALSE)
```

```
big = read.csv('bigDrugCellLine.csv')
dim(big)
head(big[, 1:5])
sum(is.na(big))
sum(duplicated(big))
```

DATA ANALYSIS

Predictive Modelling

Choice of models:

Given the high number of features (~1750) and high level of correlation between features as demonstrated by the correlation heatmap above, I decided to pick 2 models that are thought to have built-in feature selection

capability and that relatively more interpretable: LASSO and Random Forest. The better model is selected by RMSE (root mean squared error) for later analysis.

Prepare the data

Load in the drug response dataset

```
big = read.csv('bigDrugCellLine.csv')
```

Normalize the dataset.

```
subBig = big[ ,3:1749]
mean = apply(subBig, 2, mean)
std = apply(subBig, 2, sd)
subBig = as.data.frame(scale(subBig, center = mean, scale = std))
subBig$AUC = big$AUC
```

Make train and test subsets.

```
train = subBig[1: 42000, ]
test = subBig[42000: 49536, ]
train.x = subBig[1:42000, 2:1747]
train.y = subBig[1:42000, 1]
test.x = subBig[42000:49536, 2:1747]
test.y = subBig[42000:49536, 1]
```

Train a Lasso model

Select the best *lambda* for LASSO from 3-fold cross validation in the training set.

```
library(glmnet)
lasso.cv = cv.glmnet(train.x, train.y, alpha = 1, nfolds = 3)
```

Retrain and test the model on testing set

```
model = glmnet(train.x, train.y, family = 'gaussian', alpha = 1)
elasticPredict = predict(model, s = lasso.cv$lambda.min, newx = test.x, type = 'response')
lasso.RMSE = sqrt(mean((elasticPredict - test.y)^2))
```

Get test set RMSE

```
lasso.RMSE = readRDS('lasso.RMSE.RData')
lasso.RMSE
```

```
## [1] 1.296325
```

Train a Random Forest model

Grid search was used to select the `mtry` and `nodesize` hyperparameters for the model. Tuning was done on the training set. Due to limited computational power, only 3 values were tested for each.

```
library(randomForest)
set.seed(101)

mtry = c(32, 64, 128)
nodesize = c(32, 64, 128)
RF.all.scores = matrix(rep(1, 9), nrow = 3)
for (i in c(1:3)){
  for (j in c(1:3)){
    cat('Processing mtry ', mtry, 'nodesize ', nodesize, '\n')
```

```
    val.data = train.x[30001:42000, ]
    val.response = train.y[30001:42000, ]


    partial.train.data = train.x[1:30000, ]
    partial.train.response = train.y[1:30000, ]

    model = randomForest(partial.train.data, partial.train.response, mtry = mtry[i], nodesize = nodesiz
    RFPredict = predict(model, newdata = val.data)
    RMSE = sqrt(mean((RFPredict - val.response)^2))
    RF.all.scores[i,j] = RMSE
  }
}
```

```
RF.all.scores = readRDS('RF.all.scores.RData')
colnames(RF.all.scores) = c('nodesize 32', 'nodesize 64', 'nodesize 128')
rownames(RF.all.scores) = c('mtry 32', 'mtry 64', 'mtry 128')
RF.all.scores
```

```
##          nodesize 32 nodesize 64 nodesize 128
## mtry 32     1.181161    1.185261     1.193648
## mtry 64     1.177684    1.180305     1.186900
## mtry 128    1.177133    1.179815     1.184394
```

The combination (mtry, nodesize) = (128, 32) were the got the best cross-validated RMSE among tested hyperparameter combinations and will be selected for traing the final model on test set.

```
library(randomForest)
RF.model = randomForest(train.x, train.y, mtry = 128, nodesize = 32, importance = TRUE)
RFPredict = predict(RF.model, newdata = test.x)
RF.RMSE = sqrt(mean((RFPredict - test.y)^2))
```

Get test set RMSE

```
RF.RMSE = readRDS('RF.RMSE.RData')
RF.RMSE
```

```
## [1] 1.201214
```

Calculate R^2 value

```
RFPredict = readRDS('RFPredict.RData')
SS.tot = sum((test.y - mean(test.y))^2)
SS.res = sum((test.y - RFPredict)^2)
R2 = 1 - SS.res/SS.tot
R2
```

```
## [1] 0.6873981
```

The Random Forest model outperformed the LASSO model based on test RMSE score. This is reasonable, given that biological processes governing cancer cell survival are highly complex and are consist of many non-linear relationships among proteins and other molecules inside the cellular environment.

Model Interpretation

The Random Forest model can be used not only to predict drug response, but also to further understand what features from the drugs and the cell lines are good predictors. This knowledge would be very helpful for designing new drugs as well as identifying new proteins that are potential drug targets.

```
RF.all.importance = readRDS('RF.all.importance.RData')
```

```
globalFeatureOrder = order(RF.all.importance, decreasing = TRUE)
globalImportantMorphos = names(train.x)[globalFeatureOrder[globalFeatureOrder < 1533][1:10]]
globalImportantMorphos
```

```
##  [1] "Cytoplasm_AreaShape_Zernike_5_3"
##  [2] "Cytoplasm_Granularity_6_AGP"
##  [3] "Nuclei_Texture_SumEntropy_ER_10_0"
##  [4] "Cells_Granularity_6_AGP"
##  [5] "Nuclei_Texture_SumAverage_AGP_3_0"
##  [6] "Nuclei_RadialDistribution_MeanFrac_Mito_4of4"
##  [7] "Nuclei_Texture_Contrast_Mito_10_0"
##  [8] "Nuclei_Texture_InverseDifferenceMoment_Mito_5_0"
##  [9] "Nuclei_Texture_Entropy_ER_5_0"
## [10] "Nuclei_Texture_DifferenceEntropy_Mito_10_0"
```

```
globalImportantProteins = names(train.x)[globalFeatureOrder[globalFeatureOrder >= 1533][1:10]]
globalImportantProteins
```

```
##  [1] "YAP_pS127_Caution"      "YAP_Caution"
##  [3] "VAV1_Caution"           "alpha.Catenin"
##  [5] "CD49b"                  "beta.Catenin"
##  [7] "VEGFR2"                 "EGFR"
##  [9] "p62.Lck.ligand_Caution" "Caveolin.1"
```

The top 10 globally most important drug morphological features are not very revealing. The top 10 globally most important proteins are common in many cancer types, and thus are not very informative neither.

It has been shown in the 'Exploring the Data Sets' section that the drugs and cell lines in this study could be grouped into different clusters based on morphological profiles and protein expression levels, respectively. This suggested that doing model interpretation on the randomForest model may only reveal global predictors, while in reality for each drug cluster - cell line pair, there may be specific predictors for that pair. For this reason, refitting random forest models for observations from each of these drug cluster - cell line cluster pairs and interpreting this 'sub-model' may yield more valuable. Due to time limitation, I decided to carry out this approach with one pair: drug cluster #1 and cell line cluster #2.

Making the merged AUC file for drug cluster #1 and cell line cluster #2

```
AUCs.drug.one = merge(AUCs, drugs[drugClusters$cluster == 1, ])
AUCs.drug.one.cell.two = merge(AUCs.drug.one, cells[cellClusters$cluster == 2, ])
```

Train-test split

```
train.x.onetwo = AUCs.drug.one.cell.two[1:8000, 4:1749]
train.y.onetwo = AUCs.drug.one.cell.two[1:8000, 3]
test.x.onetwo = AUCs.drug.one.cell.two[8000:9154, 4: 1749]
test.y.onetwo = AUCs.drug.one.cell.two[8000:9154, 3]
```

Refit the model to this subset of data

```
library(randomForest)

RF.onetwo = randomForest(train.x.onetwo,
                         train.y.onetwo,
                         mtry = 128,
                         nodesize = 32,
                         importance = TRUE)
```

```
RFPredict.onetwo = predict(RF.onetwo, newdata = test.x.onetwo)
RMSE.onetwo = sqrt(mean((RFPredict.onetwo - test.y.onetwo)^2))
```

```
RMSE.onetwo = readRDS('RMSE.onetwo.RData')
RMSE.onetwo
```

```
## [1] 1.00214
```

Finding the 10 most important variables for drugs and cell lines

```
RF.onetwo.importance = RF.onetwo$importance[, '%IncMSE']
```

```
RF.onetwo.importance = readRDS('RF.onetwo.importance.csv')
featureOrder = order(RF.onetwo.importance, decreasing = TRUE)
```

```
importantMorphos = names(train.x)[featureOrder[featureOrder < 1533][1:10]]
importantMorphos
```

```
##  [1] "Nuclei_Correlation_RWC_RNA_Mito"
##  [2] "Nuclei_Correlation_RWC_Mito_RNA"
##  [3] "Cytoplasm_Texture_InfoMeas1_Mito_10_0"
##  [4] "Cytoplasm_Texture_InfoMeas2_Mito_10_0"
##  [5] "Cytoplasm_Texture_InfoMeas2_Mito_3_0"
##  [6] "Nuclei_Texture_DifferenceEntropy_DNA_3_0"
##  [7] "Cytoplasm_Granularity_8_Mito"
##  [8] "Nuclei_Correlation_Correlation_Mito_RNA"
##  [9] "Nuclei_Texture_Contrast_DNA_3_0"
## [10] "Cytoplasm_Texture_InfoMeas2_Mito_5_0"
```

```
importantProteins = names(train.x)[featureOrder[featureOrder >= 1533][1:10]]
importantProteins
```

```
##  [1] "beta.Actin_Caution"        "Syk"
##  [3] "MYH11"                     "FOXO3a_pS318_S321_Caution"
##  [5] "p70S6K"                    "CD49b"
##  [7] "MEK1_pS217_S221"           "PRDX1"
##  [9] "Chk1_pS345_Caution"        "JAK2"
```

It is interesting to see that the most important morphological features of the drugs in this cluster are related to the mitochondria RNAs. Mitochondria is the organelle that generate energy for the cell, and it contains in itself a second set of DNA that instructs protein synthesis (the first and more well-known set of DNA is located within the cell nuclei). Since cancer is basically a DNA disease, damages in mitochondria DNA and RNAs could be just as harmful as damages in nuclei DNA. It is reasonable to hypothesize that drugs in cluster #1 exert their effects on proteins that regulate mitochondria DNA or RNA activity in order to induce response. This hypothesis is corroborated by the identities of the top 10 most important proteins above. β-actin have been reported to regulate mitochondria DNA transcription 8. SYK (spleen tyrosine-kinase) is a tyrosine-kinase that is involved in the mitochondria respiratory chain 9 and have been reported to play the dual-role of both tumor promoter and tumor suppressor in various cancers 10. p70S6K is an important serine-threonine kinase in the mTOR pathway that controls mitochondria activity and is frequently activated in various cancers 11. In short, looking at this one specific drug cluster - cell line cluster pair suggested potential mode of actions of the drugs inside drug cluster #1 and provided a list of promising protein targets for killing cancer cells in cell line cluster #2.

CONCLUSION AND FUTURE DIRECTION

This study explored the use of drugs' morphological profile as a new, useful set of features for building predictive models of drug response in cancer cell line. The model achived an RMSE of 1.201 and an R^2 value of 0.687. Furthermore, morphological features, when examined in parallel with -omics data from cancer cell

line, revealed potential relationships between the drugs and intracellular molecules and pathways that can be validated and exploited for development of more potent and personalized cancer therapeutics.

Some future directions that I plan to pursue further are: * Try fitting Random Forest with features selected from the LASSO. Pre-modeling feature selection might improve the model's predictive power. * Ensemble morphology - protein model with models using other data sources (e.g. morphology - RNA expression or morphology - somatic mutation) * Extensive mining of morphology - protein relationships in remaining drug cluster - cell line cluster pairs. * Validation of these pathways. * Using deep neural network (e.g. convolution neural network) to directly extract morphological features for drugs from microscopic images of standard cancer cells treated with those drugs. At the moment these features come from hand-crafted, hard-defined rules. These rules may not be able to capture all the information in the images. This would require tremendous computing power and powerful algorithms, which is a major challenge.