STAT432 FALL 2021 Final Project Report

Author: Yijun Zhao (yijunz3), Peiyi Chen (peiyic2), Rongxin Ni (rni4)

1. Project Description and Summary

According to the American Cancer Society (ACS), breast cancer is the most common cancer in American women. Breast cancer also accounts for roughly a third of all cases of cancer. In a woman's life, she has about a 12% chance to get breast cancer, and a 2.6% chance to die because of it. Breast cancer exacts a considerable amount of medical resources and economic cost. Studying breast cancer can help us to better prepare for it.

Within this project, we want to learn about breast cancer subtype identifications from four relevant papers and the Kaggle BRCA Multi-Omics (TCGA) data containing 705 breast cancer samples, 1936 variables, and 5 outcomes. Our goal is to establish efficient predictive models for four of the outcomes.

First, we get a basic understanding of the dataset by univariate analysis. For continuous variables, we remove missing values and look for outliers. We also perform cube-root transformation and standardization. For categorical variables, we delete several extremely imbalanced variables that may introduce bias. Based on our knowledge from articles, we know that certain variables are important features of Invasive Lobular Breast Cancer, so we would consider keeping them in the model while removing other unnecessary variables. As a result, we have 1607 out of 1936 features left.

We build classification models to predict PR.Status by applying Linear discriminant analysis (LDA) and kmeans clustering. Both two models have accuracies of around 80%. However, kmeans clustering has a better performance with relatively lower classification error and greater sensitivity. For histological.type modeling, we have a try on Logistic Regression and Support Vector Machine. Both models work well with AUC and accuracy of more than 0.85, but their sensitivities are surprisingly low. A possible explanation for this observation might be the extreme imbalance of the outcome.

We also hope to select a small set of biomarkers that can accurately predict all four outcomes. We choose to use the Random Forest algorithm to select the most important 50 features here. By simply summing up the importance attribute of the four fitting models for different outcomes, we have a nice new 50-feature dataset whose performance even beats our former 1607-feature dataset.

In order to evaluate the new dataset, we use three-fold cross-validation with AUC and find the averaged AUC improved on this smaller dataset. Therefore, we can conclude that the selected variables can help us to better predict the four outcomes.

2. Literature Review

(1) Comprehensive Molecular Portraits of Invasive Lobular Breast Cancer [1]

This article provides a comprehensive molecular portrait of Invasive lobular carcinoma (ILC). The study identified alterations (both mutations and copy-number changes) that discriminate between IDC and ILC demonstrating that ILC is a clinically and molecularly distinct disease. Researchers discovered that PTEN inactivating alterations are relatively higher in LumA ILC compared to LumA IDC. One typical characteristic of ILC is small discohesive neoplastic cells invading the stroma in a single-file pattern led by the absence of E-cadherin (CDH1) protein expression. FOXA1 selected mutations are another noticeable feature of ILC. ILC also has subtypes termed reactive-like, immune-related, and proliferative, and their tumor proliferation was generally lower than that of IDC.

(2) Identification of different subtypes of breast cancer using tissue microarray [2]

Breast cancer is a heterogeneous disease, in other words, it may have various symptoms and responses to therapy for different patients. One possible explanation for these experiences is the origin of neoplastic cells. Breast cancer can be classified into pathologically distinct subtypes based on gene expressions. The subtypes include expressing ER (ER+) luminal and non expressing ER (ER-) tumors which can be divided into HER2+, basal-like, and normal-like subgroups. There exists a significant correlation between subtypes with ethnicity and histological grade, but they are not associated with age, menopausal status, tumor size, or lymph nodes status. Luminal A is the most commonly diagnosed subtype which is majorly observed in over 50-year-old patients.

(3) Supervised Risk Predictor of Breast Cancer Based on Intrinsic Subtypes [3]

It's really hard to classify breast tumors since breast cancer is a heterogeneous disease. The multi-gene prediction system can provide more accurate prognostic information and provide a more reliable reference for selecting treatment options, which is an important breakthrough direction for tumor precision treatment. The researchers developed a 50-gene subtype using microarray and quantitative reverse transcriptase-polymerase chain reaction data from 189 prototype samples. The results show that intrinsic subtyping at diagnosis can provide significant prognostic and predictive information to standard parameters for patients with breast cancer.

(4) Breast cancer and associated factors: a review [4]

Breast cancer is a common disease that can occur in women and rarely in men. Patients face many psychological problems like stress and depression. There are many Breast cancer risk factors, some are related to people's lift style like cigarette-smoking, alcohol use, diet, and exercise are changeable. Some types of breast cancer are affected by hormones, but the development and progress are not clear. In conclusion, factors as aging, history of breast cancer development in the family, certain changes in breasts, genetic changes, history of productivity

and menopause, lack of physical activity, alcohol use, diet and nutrition, race, and radiation therapy to chest are risk factors of breast cancer.

3. Summary Statistics and data processing

The dataset we're working on contains 705 breast cancer samples and 1941 variables in total. Except for the vital.status variable we discard and 5 outcomes, there are 1936 usable variables including 860 copy number variations (cn), 249 mutations(mu), 604 gene expressions(rs), and 223 protein levels (pp).

We preprocess the data before performing further analysis. First, we separate outcomes from the data, which gives us two datasets `data_x` and `data_y`. After reading the description and going through the data in hand, we find that variables in the types of rs and pp are continuous, while the rest are categorical.

According to the article "Comprehensive Molecular Portraits of Invasive Lobular Breast Cancer", we know that mutations PTEN, TBX3, and FOXA1 are ILC enriched features, so these variables should definitely be included in our dataset. Moreover, the article studies a list of recurrently mutated genes in breast cancer, which should also be considered. We can remove other `mu` variables that are not mentioned, since they may not provide as much useful information as the previous ones.

In order to manipulate two kinds of variables separately, we split the dataset `data_x` into two small datasets `data_continous` and `data_cat`, one of which contains all the continuous variables while the other includes the categorical variables.

Next, we look at the distributions of continuous variables by using `describe()` function in R. If skewness is between -0.5 and 0.5, we consider the distribution to be approximately symmetric; if the absolute value is between 0.5 and 1, we consider the distribution to be moderately skewed. Otherwise, if the absolute value of skewness is greater than 1, we consider the distribution to be highly skewed.

```r
#Check how many continuous variables are skewed distributed
summary_continous = describe(data_continous)
sum(abs(summary_continous$skew) >= 1)
```

```
[1] 195
```

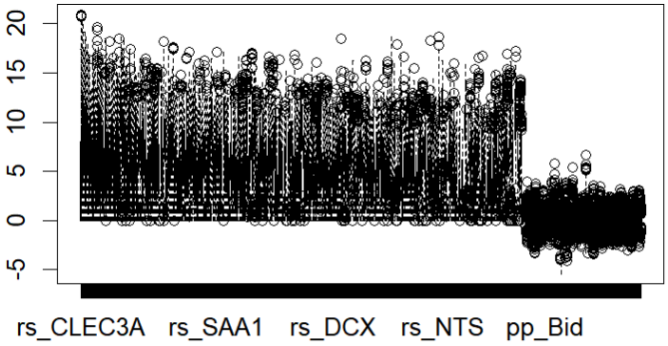
We find 195 variables with skewness whose absolute values are greater than 1. Since these highly skewed variables contain both positive, zero, and negative values, we would perform a cube-root transformation.

```{r}
list_skew = which(abs(summary_continous$skew) >= 1)
data_continous[,list_skew] = data_continous[,list_skew]^(1/3)
#View(data_continous)
sum(abs(describe(data_continous)$skew) >= 1)
```

```
[1] 26
```

Observe that 26 variables are still highly skewed after the transformation, so we want to remove them. The new dataset `data_continous` contains 705 observations and 801 variables.

Then we remove missing values from the dataset, also check for outliers by drawing boxplots and using the function `boxplot(data_continous)$out`.



Boxplots of the dataset of continuous variables

We find 5823 outliers in total. However, since those outliers only take roughly 1% of all the values, and we do not observe outliers that are extremely far in the boxplots, we think these outliers are more likely to come from data variation instead of typo or incorrect measurement, we decide not to drop them. We standardize our data to make variables comparable.

```{r}
data_continous <- scale(data_continous, center = TRUE, scale =
TRUE)
```

The next step is to work with categorical data. Since covariate imbalance may introduce bias into our model, we want to remove those extremely imbalanced covariates. Therefore, we look at the

variables in the type "mu". Since they only have two classes 0 and 1, we will look for the ratio of these two classes in each variable, and then remove the variables that have very high ratios.

We do not have a specific standard for “very high”, so we set it to be 25 initially. However, we realize that more than three of the fourths of variables will be removed, including some relatively important ones such as “mu_PTEN”, “mu_FOXA1”, etc. After several attempts, we decide the ratio to be 45 such that we can eliminate variables without losing much important information. We remove categorical variables "mu_TP53BP1", "mu_ERBB2", and "mu_RPGR".

Lastly, we combine manipulated datasets `data_continous` and `data_cat` into a new dataset with 705 observations and 1607 variables. This is the dataset we’re going to use in modeling.

4. Modeling PR.status

We choose to use Linear discriminant analysis (LDA) and kmeans clustering for PR.Status modeling. We split the whole dataset into training and testing datasets with a ratio of 75%: 25%. Next, we preprocess the dataset by its output value: there exist 122 missing values that should be removed, classes "Indeterminate", "Not Performed" and "Performed but Not Available" contain few values, so we will only focus on classes "Positive" and "Negative".

We perform linear discriminant analysis by applying the `lda()` function. We then predict the results of testing data and get the classification error to be 0.2083333.

```r
dig.lda = lda(pr_data_train$x,pr_data_train$y)

  # Use classification error as the evaluation criterion.
Ytest.pred = predict(dig.lda, pr_data_test$x)
mean(pr_data_test$y != Ytest.pred$class)

  # You need to provide sufficient information (table, figure and descriptions) to
demonstrate the model fitting results
table(Ytest.pred$class, pr_data_test$y)
```

Code of linear discriminant analysis for PR.Status

In order to perform kmeans clustering, we use the function kmeans() and set the number of clusters to be 2. We compare the actual classification in the dataset with the kmean clustering result and obtain a classification error of 0.1691542.

```r
set.seed(12345)
pr_mat_train = cbind(pr_data_train$x,pr_data_train$y)
kmeanfit <- kmeans(pr_mat_train[,-dim(pr_mat_train)[2]],2)


  # Use classification error as the evaluation criterion.
mean((pr_mat_train$`pr_data_train$y` + 1) != kmeanfit$cluster)

  # You need to provide sufficient information (table, figure and descriptions) to
demonstrate the model fitting results
table(kmeanfit$cluster, pr_mat_train$`pr_data_train$y`)
```

Code of kmeans for PR.Status

Both two approaches can reach an accuracy of around 80%, which are relatively good results for our understanding. We further reported all the confusion matrix, sensitivity, and specificity as below.

<table>
<tr><td></td><td>LDA</td><td>kmeans</td></tr>
<tr><td>Confusion Matrix</td><td><table><tr><td></td><td>0</td><td>1</td></tr><tr><td>False</td><td>38</td><td>17</td></tr><tr><td>True</td><td>13</td><td>76</td></tr></table></td><td><table><tr><td></td><td>0</td><td>1</td></tr><tr><td>False</td><td>78</td><td>4</td></tr><tr><td>True</td><td>64</td><td>256</td></tr></table></td></tr>
<tr><td>Classification Error</td><td>0.2083333</td><td>0.1691542</td></tr>
<tr><td>Accuracy</td><td>0.79166667</td><td>0.83084577</td></tr>
<tr><td>Sensitivity</td><td>0.8172043</td><td>0.98461538</td></tr>
<tr><td>Specificity</td><td>0.74509804</td><td>0.54929577</td></tr>
</table>

Table of the PR.Status modeling result

We have a sensitivity of 0.8172043 for linear discriminant analysis and sensitivity of 0.98461538 for kmeans, but have a specificity of 0.74509804 for linear discriminant analysis and specificity of only 0.54929577 for kmeans, which leads to the overall accuracy of 0.79166667 and 0.83084577 for linear discriminant analysis and kmeans.

The performance of kmeans clustering is better than linear discriminant analysis with higher accuracy, one possible reason is that we use different datasets for prediction. Since kmeans clustering does not require a train-test split, we simply use the training data to predict while the testing dataset is used for prediction in LDA.

Comparing the two approaches, kmeans is more sensitive in terms of telling positive PR Status, but it has a poor performance in identifying “PR-Negative” patients. The LDA model is more conservative in terms of concluding any positive, but it has greater specificity.

5. Modeling histological.type

We choose to use Logistic Regression and Support Vector Machine for histological.type modeling. The same as the PR.status part, in order to test the model performance, we split the whole dataset into training and test datasets with a ratio of 75% : 25%. We then preprocess the dataset by its output value similar to the last part: only consider “infiltrating lobular carcinoma”

and “infiltrating ductal carcinoma”, and eliminate all other classes of the outcome. But we find all the data points’ histological.type outcomes are either “infiltrating lobular carcinoma” or “infiltrating ductal carcinoma”, so we actually used all the data points we have for histological.type modeling.

For Logistic Regression, we use the glmnet function in the glmnet library, with family=“binomial” and alpha=0 to model the 75% training data and treat their histological.type as the outcome y value. By predicting the results on the remaining 25% test dataset, we got an AUC of 0.8961207 and an accuracy of 0.8926554 (more detailed results like confusion matrix are reported in the table of the histological.type modeling result).

```
library(glmnet)
logistic.fit <- glmnet(hist_data_train$x, hist_data_train$y, alpha = 0, family = "binomial")
pred = predict(logistic.fit, data.matrix(hist_data_test$x), type = "response", s=min(fit$lambda))

# Use AUC as the evaluation criterion.
library(ROCR)
roc <- prediction(pred, hist_data_test$y)
performance(roc, measure = "auc")@y.values[[1]]
```

Code of Logistic Regression for histological.type

For Support Vector Machine, we use the svm function in e1071 library, with type='C-classification', kernel='linear', scale=FALSE, and cost = 1 for modeling the 75% training data. To extract the probability attribute from the prediction result, we got an AUC of 0.881681 and an accuracy of 0.9096045 (more detailed results like confusion matrix are reported in the table of the histological.type modeling result).

```
library(e1071)
svm.fit <- svm(as.factor(hist_data_train$y) ~ .,
               data = data.frame(hist_data_train$x),
               type='C-classification',
               probability = TRUE,
               kernel='linear', scale=FALSE, cost = 1)

pred = predict(svm.fit, hist_data_test$x, probability=TRUE)
pred_prob = attr(pred, "probabilities")
```

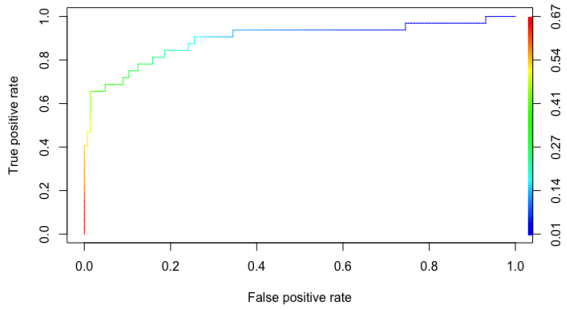
Code of SVM for histological.type

Both logistic regression and SVM can reach AUC and accuracy of more than 0.85 for histological.type, which are very good results for our understanding. We further reported all the AUC, confusion matrix, accuracy, sensitivity, specificity, and the ROC plot results as below. As we can see from the ROC plot, both methods achieve very good results but the logistic regression one covers more area, which can also be shown from the AUC value results. From the confusion matrix, we can clearly see that the histological.type distribution is very unbalanced. More than 80% of data points have ‘False’ results, which leads to poor performance on ‘True’

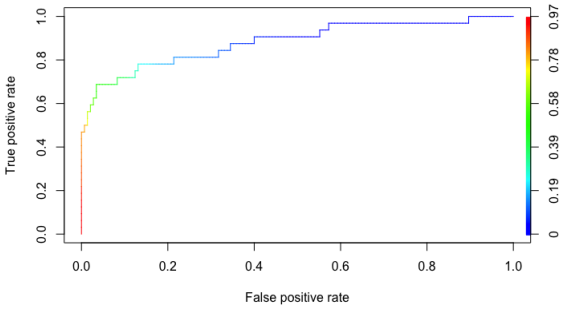
result determination for both logistic regression and SVM. We have a sensitivity of only 0.4375 for logistic regression and sensitivity of 0.656 for SVM, but have a specificity of as high as 0.9931034483 for logistic regression and specificity of 0.97241379 for SVM, which leads to the overall accuracy of 0.8926554 and 0.898305 for logistic regression and SVM. Thus, we concluded that even though our model cannot classify 'True' results well due to the extremely unbalanced dataset, our model still works well overall.

<table>
<tr><td></td><td>Logistic Regression</td><td>SVM</td></tr>
<tr><td>AUC</td><td>0.8954741</td><td>0.8853448</td></tr>
<tr><td>Confusion Matrix</td><td><table><tr><td></td><td>0</td><td>1</td></tr><tr><td>False</td><td>144</td><td>18</td></tr><tr><td>True</td><td>1</td><td>14</td></tr></table></td><td><table><tr><td></td><td>0</td><td>1</td></tr><tr><td>False</td><td>141</td><td>11</td></tr><tr><td>True</td><td>4</td><td>21</td></tr></table></td></tr>
<tr><td>Accuracy</td><td>0.8926554</td><td>0.898305</td></tr>
<tr><td>Sensitivity</td><td>0.4375</td><td>0.656</td></tr>
<tr><td>Specificity</td><td>0.9931034483</td><td>0.97241379</td></tr>
</table>

Table of the histological.type modeling result



ROC plot of Logistic Regression



ROC plot of SVM

6. Variable selection for all outcomes

The goal of this part is to select a small set of biomarkers (only 50 out of all 1936 variables) to accurately predict all the four outcomes (vital.status, PR.Status, ER.Status, HER2.Final.Status, and histological.type). Because we want to use cross-validation for evaluation in this part, we did

not split our whole dataset into training and test datasets as the last two modeling parts. Instead, we used all the data points we have in this part.

In order to reduce the dimensions quickly, we simply decide to use Random Forest to choose the most important 50 features. We first tune the random forest parameters by doing a 3-fold cross-validation on histological.type as the figure shows. We get the best tuning result of 0.9163121 accuracy and 0.6811635 kappa when mtry = 500 and min.node.size = 10. The best tuning parameter also gives us almost 100% accuracy on the whole histological.type dataset.

```
# tune rf parameters
library(caret)
tunegrid <- expand.grid(mtry = c(50, 100, 500, 700),min.node.size = c(1,5,10,20),
                        splitrule = "gini")
ctrl <- trainControl(method = "cv", number = 3)
rf.fit = train(y=as.factor(hist_data$y),
                   x = hist_data$x, method = 'ranger',
                   trControl = ctrl, num.trees = 400, tuneGrid = tunegrid,
                   respect.unordered.factors = "partition")
rf.fit
```

Code of random forest parameters tuning

We then apply our best tuning parameters with ntree=400, mtry=500, nodesize=10 to all the four y values by using the randomForest function in the randomForest library and get four fitting models. Summing up the importance attribute of all four fitting models, we have the importance level of each feature for all four outcomes in the dataset. Simply choosing the top 50 important features and eliminating all other ones, we have our new dataset with 50 variables only.

```
# select a total of 50 variables

rf.fit_pr = randomForest(pr_data$x, as.factor(pr_data$y), ntree = 400,
                      mtry = 500, nodesize = 10)
rf.fit_er = randomForest(er_data$x, as.factor(er_data$y), ntree = 400,
                      mtry = 500, nodesize = 10)
rf.fit_her = randomForest(her_data$x, as.factor(her_data$y), ntree = 400,
                      mtry = 500, nodesize = 10)
rf.fit_hist = randomForest(hist_data$x, as.factor(hist_data$y), ntree = 400,
                      mtry = 500, nodesize = 10)
total_importance = rf.fit_pr$importance + rf.fit_er$importance + rf.fit_her$importance + rf.fit_hist$importance
idxes = sort(total_importance, decreasing = TRUE, index.return=TRUE)$ix[1:50]
```

Code of choosing 50 features

```
 [1] "pp_ER.alpha"     "rs_PGR"          "mu_CDH1"         "cn_PPP1R1B"      "pp_beta.Catenin" "rs_CYP2B7P1"
 [7] "pp_GATA3"        "cn_PNMT"         "cn_IKZF3"        "rs_SCUBE2"       "rs_TFF1"         "rs_RGS22"
[13] "rs_ANKRD43"      "rs_FSIP1"        "rs_AZML1"        "rs_CLSTN2"       "rs_GRPR"         "rs_DEGS2"
[19] "rs_PPP1R14C"     "cn_STAC2"        "rs_SYT9"         "rs_LOC389033"    "rs_C1orf64"      "rs_WNK4"
[25] "rs_AFF3"         "rs_TTC36"        "rs_HPX"          "rs_TFF3"         "rs_SERPINA11"    "rs_FXYD1"
[31] "rs_SOX11"        "rs_NAT1"         "rs_TRH"          "rs_PNMT"         "rs_DACH1"        "rs_HS6ST3"
[37] "pp_INPP4B"       "rs_PTPRT"        "pp_S6"           "pp_TFRC"         "rs_FLJ45983"     "rs_SLC7A2"
[43] "rs_ELOVL2"       "pp_JNK2"         "rs_PRODH"        "pp_SLC1A5"       "rs_KCNJ3"        "rs_ADAMTS15"
[49] "rs_TNNT3"        "rs_KRT16"
```

The 50 features chosen by our algorithm

In order to evaluate our new dataset, we follow the three-fold cross-validation with AUC evaluation criteria stated in the requirements. Using set.seed(1); sample(1:3, 705, replace = TRUE), we naturally have three folds of data with value 1, 2 or 3. Then, by treating each 1, 2, or 3 fold-value data points as the test dataset and fitting models on the rest data points, we got three-fold cross-validation. We also calculate the AUC value for the test dataset in each fold, so just averaging them up in the end we would have the mean AUC we want for the particular outcome we choose.

```
  # evaluation criteria is based on a three-fold cross-validation with AUC for each outcome, and then average the
cross-validated AUC of all four outcomes
  set.seed(1)
  all_fold_ids = sample(1:3, 705, replace = TRUE)

  eval <- function(data_x, data_y, fold_ids) {
    auc_list = c()
    for (i in 1:3) {
      test_idxes <- which(fold_ids==i)
      test_x <- data_x[test_idxes,]
      test_y <- data_y[test_idxes]
      train_x <- data_x[-test_idxes,]
      train_y <- data_y[-test_idxes]

      fit <- glmnet(train_x, as.factor(train_y), alpha = 0, family = "binomial")
      pred = predict(fit, data.matrix(test_x), type = "response", s=min(fit$lambda))
      roc <- prediction(pred, test_y)

      auc <- performance(roc, measure = "auc")@y.values[[1]]
      auc_list = c(auc_list, auc)
    }
    return(mean(auc_list))
  }
```

Code of our cross-validation evaluation

After trying all the svm, random forest, and logistic regression as the fitting models for the final cross-validation, we got the best result on logistic regression of 0.9323858 as the following table shows. For SVM, we used the same svm function as the last part with type='C-classification', kernel='linear', scale=FALSE, and cost = 1. Even though svm works very well for histological.type and can reach AUC of as high as 0.9112326, it has very bad performance on PR.Status and ER.Status. As a result, it has a poor mean AUC result of only 0.5637454 in total. For random forest, we used the same function and parameters we used for variable selections, except that we set mtry=50 here due to much less variables we have. It works equally well for all four outcomes and can reach AUC of 0.853399 in total. For logistic regression, we also used the same function as the last part, using the glmnet function in the glmnet library with family="binomial" and alpha=0. It gives us AUC of more than 0.90 for all four outcome y values and resulted in AUC of 0.9323858 in total. We also tested the logistic regression result on our former dataset with all 1607 variables and got the average AUC of 0.903281. Surprisingly, our dataset with only 50 variables we selected by random forest beats the former dataset with 1607 variables. We can then definitely conclude that our variable selection algorithm is not only workable but also helpful for all the PR.Status, ER.Status, HER2.Final.Status, and histological.type four outcome predictions.

	svm	rf	lr	lr on 1607 features
PR.Status	0.09377239	0.841661	0.9196236	0.8840938
ER.Status	0.3454487	0.9085602	0.9582878	0.9433264
HER2.Final.Status	0.9045281	0.8403978	0.9144527	0.8815358
histological.type	0.9112326	0.8229771	0.9371789	0.904168
Total	0.5637454	0.853399	0.9323858	0.903281

Table of prediction with 50 features
(svm: support vector machine; rf: random forest; lr: logistic regression)

Reference:

[1] https://www.cell.com/cell/fulltext/S0092-8674(15)01195-2
[2] https://pubmed.ncbi.nlm.nih.gov/21655659/
[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2667820/
[4] https://pubmed.ncbi.nlm.nih.gov/28316699/