Structural protein sequences

wasu wisanuyothin

19/6/2563

1. Introduction

Protein data bank (pdb) archive is a place that biologist store atomic coordinate and other important information that describe the protein and other significant biological macromolecule. To visualize protein, biologist usually turn protein into crystallized form. Biologist use techniques such as X-ray diffraction, neutron diffraction, powder diffraction, electron crystallography, electron microscopy to determine the location of atom in protein relative to others atom in the molecule. The function of each protein is catagorize into classificattion such as hydrolase, transferase, oxidoreductase and ligase. The result observed from crystallization are residue count, resolution, molecular weight, density and pH. The definition of these result will be further explain in section 1.1 dataset.

The goals of this project are to visuallize distribution of different observations from crystalization across different classification of protein and build classification model to classify function of protein.

1.1 Dataset

The data we used is from kaggle[1], but the original data is retrived from Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank (PDB)[2]. The data have more than 141,000 proteins. The column that we used in this analysis are experiment techniques, residue count, resolution, molecular weight, density matthews, density percent sol and pH. The explanation of the columns are as follows:
1. Experiment techniques: The techniques that use to visuallize the crystal of protein and determine the location of atoms.
2. Residue count: Number of amino acid consist in the protein.[3]
3. Resolution: The distance corresponding to the smallest observable feature. High number of resolution mean lower quality of strucure. For example, 1 A(angstrom) resolution mean you can distinguish structure that are apart from each other 1 A. If resolution are 4 A, you can only seperate two structure if they are located far from each other than 4 A. Thus, the lower resolution mean better quality.[4]
4. Molecular weight: Molecualr weight of the protein.
5. Density matthews: The density of the crystal, expressed as the ratio of the volume of the asymmetric unit to the molecular mass of a monomer of the structure.[5] Its associate with telling the macrostructure.[6]
6. Density percent sol: Density value calculated from the crystal cell dimensions and content, expressed as percent solvent.[7]
7. pH: The pH value at which the crystal was grown.[8]

[1] https://www.kaggle.com/shahir/protein-data-set
[2] http://www.rcsb.org/pdb/
[3] https://www3.cmbi.umcn.nl/wiki/index.php/Residue_number#:~:text=The%20average%20protein%20contains%20multiple,the%20twenty%2
[4] https://proteopedia.org/wiki/index.php/Resolution#:~:text=In%20X%2Dray%20crystallography%2C%20resolution,1.5%20%C3%85%2C%20
[5] https://www.rcsb.org/pdb/staticHelp.do?p=help/advancedsearch/crystalProperties.html.
[6] https://people.mbi.ucla.edu/sawaya/tutorials/Characterize/characterize.html
[7] https://www.rcsb.org/pdb/staticHelp.do?p=help/advancedsearch/crystalProperties.html.
[8] https://www.rcsb.org/pdb/staticHelp.do?p=help/advancedsearch/crystalProperties.html.

1.2 Model evaluation

We will evaluate our model by using Overall accuracy that get from confusionMatrix() function. The overall accuracy is calculated from:

$$\text{Overall accuray} = \frac{\text{Number of sample that predicted right}}{\text{Number of all sample}}$$

1.3 Process of study

The 4 main processes of this study are:
1. Data preparation: Downloading data, joining 2 dataset, filter out column that we didn't use and remove NA values.
2. Data visualization: View the distribution of data and find correlation between variables.
3. Model creation: Create model to predict the classification of protein.
4. evaluation: Evaluate model using overall accuracy
5. Summarize and report: find the best model from evaluation and report

2. Method and Analysis

2.1 Data preparation

In this part, we are going to download the dataset and required packages. The name of dataset that we download are pdb_data_no_dups and pdb_data_seq. Then we join the 2 dataset into one dataset name pdb data. The pdb_data_seq have to filter out some rows that are dublicate after filter column chainId. The column that we are not going to use in this project is crystallizationMethod, crystallizationTempK, pdbxDetails, publicationYear and structureId due to lack of usage and high amount of NAs in some column. The NAs in other column will be drop as well. The last feature that we are going to filter is pHvalue. We know that in most circumstances, pH values range in 0 to 14. There are a few data point that contain pH value more than 14, so we are going to filter it out. The code that doing what describing before are shown below:

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(dslabs)) install.packages("dslabs")
if(!require(corrplot)) install.packages("corrplot")
if(!require(randomForest)) install.packages("randomForest")
#Structural Protein Sequences
#https://www.kaggle.com/shahir/protein-data-set

#download pdb_data_no_dups
dl <- tempfile()
download.file("https://raw.githubusercontent.com/wasuwis/EDX-capstone-2/master/pdb_data_no_dups.zip",dl
pdb_data_no_dups <- read_csv(unzip(dl))

#download pdb_data_seq
dl_2 <- tempfile()
download.file("https://raw.githubusercontent.com/wasuwis/EDX-capstone-2/master/pdb_data_seq.zip",dl_2)
pdb_data_seq <- read_csv(unzip(dl_2))

#exclude chainid column because we dont use it in this project
```

```
pdb_data_seq <- pdb_data_seq %>%
  subset(select = -c(chainId)) %>%
  distinct()

#combine pdb_data_no_dubs with pdb__dat_seq
pdb <- left_join(pdb_data_no_dups,pdb_data_seq)

#exclude the following column due to lack of usesage and large amount of NAs
pdb <- pdb %>% subset(select=-c(crystallizationMethod,
                                crystallizationTempK,pdbxDetails,
                                publicationYear,structureId)) %>%
  drop_na()

pdb <- as.data.frame(pdb)

# clear out phvalue more than 14
pdb <- pdb %>% filter(phValue<=14)
```

2.2 Data visualization

First we will look at structure of pdb data. We will see that we have 154,917 rows as number of sample and 10 columns that consisted of classification, experimental technique, Type of macromolecule, residue count, resolution, molecular weight, density matthews, density percent sol, pH value and sequence.

```
##  classification     experimentalTechnique macromoleculeType   residueCount
##  Length:154917      Length:154917         Length:154917      Min.   :    2
##  Class :character   Class :character      Class :character   1st Qu.:  316
##  Mode  :character   Mode  :character      Mode  :character   Median :  590
##                                                              Mean   : 2985
##                                                              3rd Qu.: 1448
##                                                              Max.   :89160
##    resolution     structureMolecularWeight densityMatthews  densityPercentSol
##  Min.   : 0.480   Min.   :     489         Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 1.860   1st Qu.:   36659         1st Qu.: 2.290   1st Qu.:46.14
##  Median : 2.250   Median :   68085         Median : 2.650   Median :53.46
##  Mean   : 2.362   Mean   :  522966         Mean   : 2.843   Mean   :53.90
##  3rd Qu.: 2.800   3rd Qu.:  169928         3rd Qu.: 3.200   3rd Qu.:61.41
##  Max.   :11.500   Max.   :97730536         Max.   :12.700   Max.   :90.33
##     phValue         sequence
##  Min.   : 0.000   Length:154917
##  1st Qu.: 6.000   Class :character
##  Median : 7.000   Mode  :character
##  Mean   : 6.807
##  3rd Qu.: 7.500
##  Max.   :12.000
```

2.2.1 Techniques

The Most use techniques in our dataset is X-ray diffraction. This techniques accounted for 99.9 percent of all sample in our data.

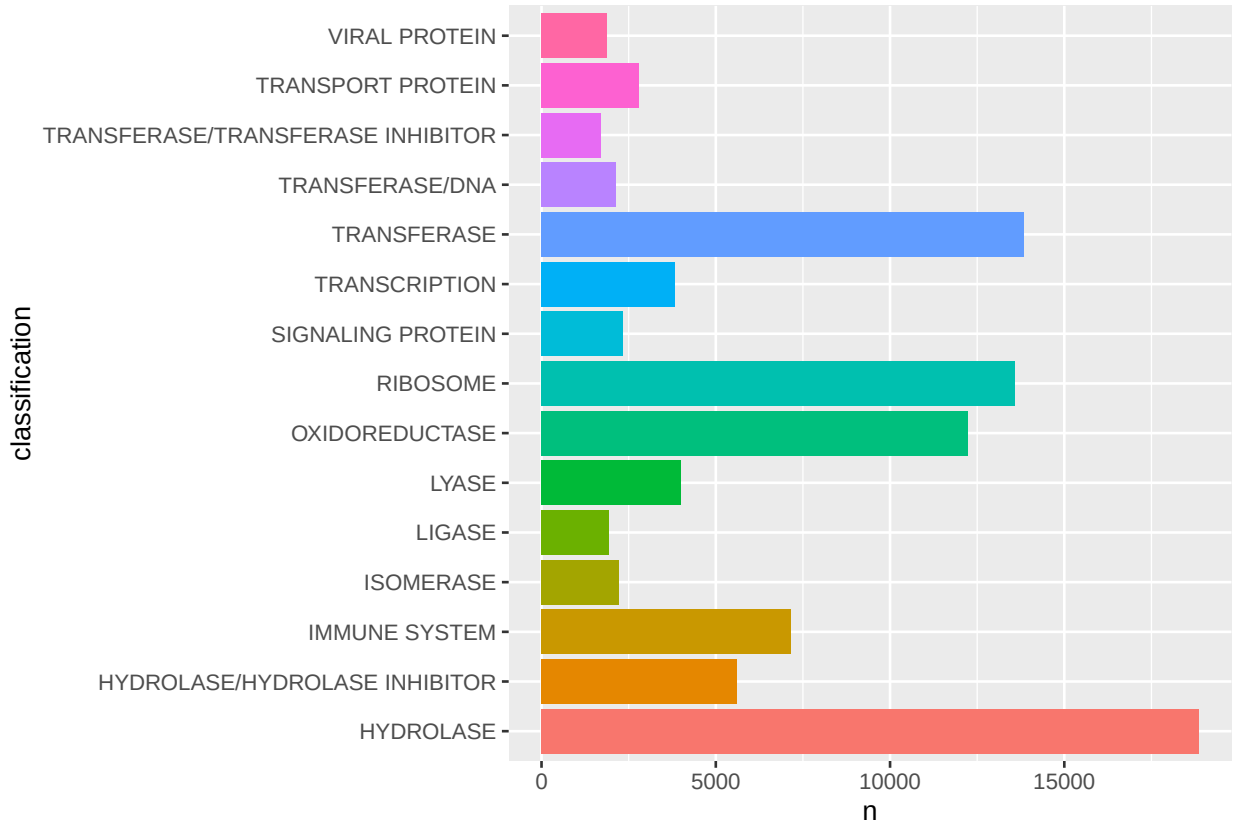
```
## # A tibble: 10 x 2
```

```
##    experimentalTechnique                       n
##    <chr>                                   <int>
##  1 X-RAY DIFFRACTION                      154869
##  2 NEUTRON DIFFRACTION                        22
##  3 ELECTRON CRYSTALLOGRAPHY                    6
##  4 POWDER DIFFRACTION                          6
##  5 NEUTRON DIFFRACTION, X-RAY DIFFRACTION      5
##  6 X-RAY DIFFRACTION, EPR                      5
##  7 ELECTRON MICROSCOPY                         1
##  8 EPR, X-RAY DIFFRACTION                      1
##  9 SOLUTION SCATTERING, X-RAY DIFFRACTION      1
## 10 X-RAY DIFFRACTION, NEUTRON DIFFRACTION      1
```

2.2.2 Classification

When we look into classification among all sample. We see that there are 3,693 classifications. To make our model that we are going to create, we can't build it to classify 3,693 class due to long run time and need of high spec computer. We are going to build the model for the top 15 classification. When we look at the total number of sample in the top 15 classification, we see that it accounted for 60 percents of data from all 3,693 classification. The number of sample in each top 15 classification are shown below:

Fig. 1 Number of protein in top 15 classification



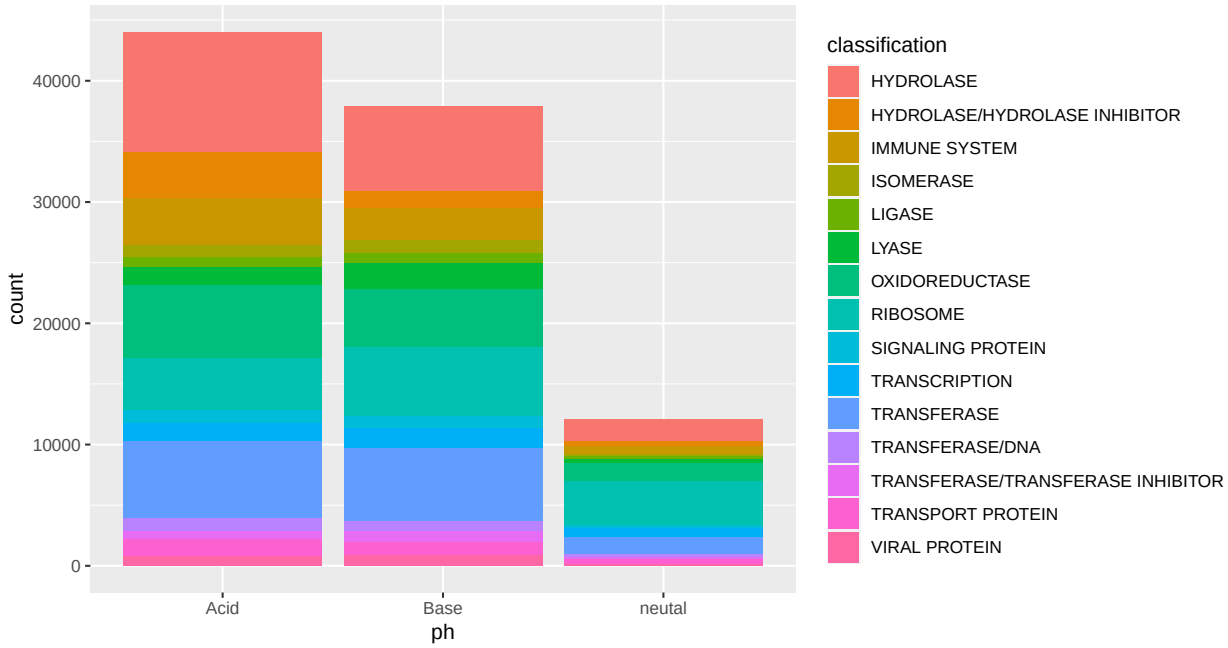
2.2.3 pH

From our knowledge, we know that substance that have pH higher than 7 is base, pH lower than 7 is acid and if pH equal 7, it is neutral. We use ph_test() function that we create to interprete pH value. The code for function are shown below.

```
#pH interpretation
ph_test <- function(phValue){

  if(phValue<7){return("Acid")}

  if(phValue>7){return("Base")}

  if(phValue==7){return("neutal")}

  }
```

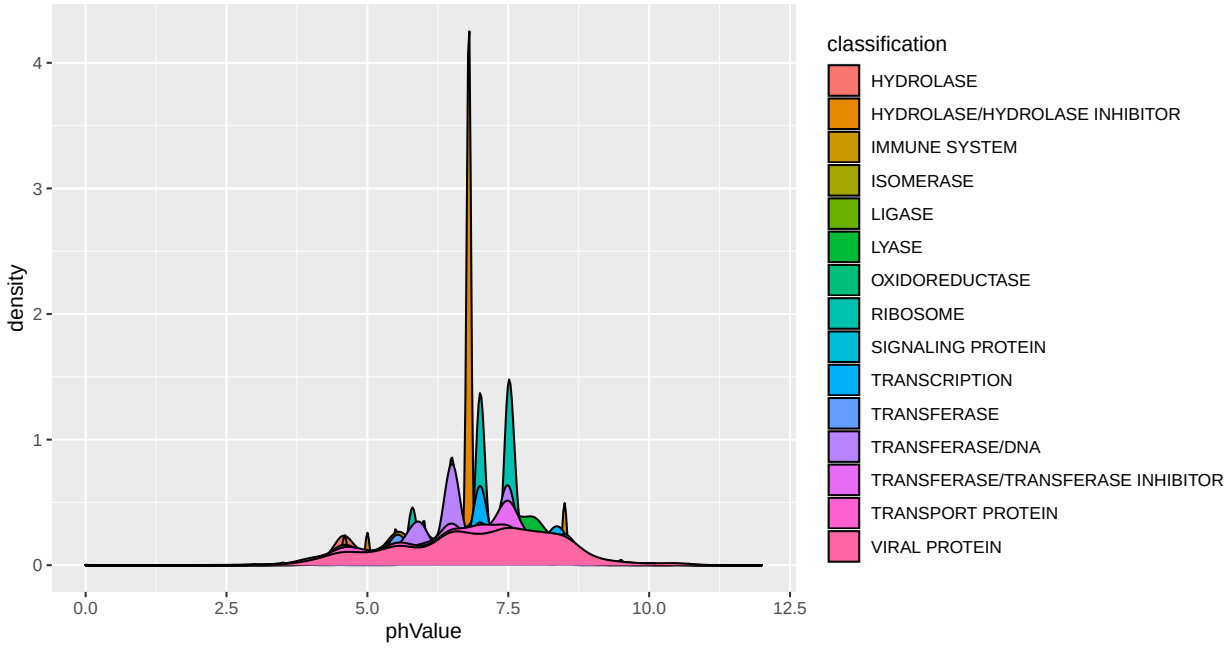
As we can see from figure 2, the amount of protein in acid and base type are about the same with a small amount of neutral type. If we look at the top 15 classification, we can see that every classification tend to have similar distribution in every type except Hydrolase/Hydrolase inhibitor class are largly acidic.

Fig. 2 Distribution of top 15 classification in acid, base and neutral pH



In the top 15 classification we can see that the distribution of pH values are varies and don't seem to be different among different classification. We can see from Figure 3 that Hydrolase/Hydrolase inhibitor seem to have the most similar pH of sample in their own classification, resulted in highest density in that pH range. The median of pH of Hydrolase/Hydrolaser inhibitor is 6.8, which is acidic.

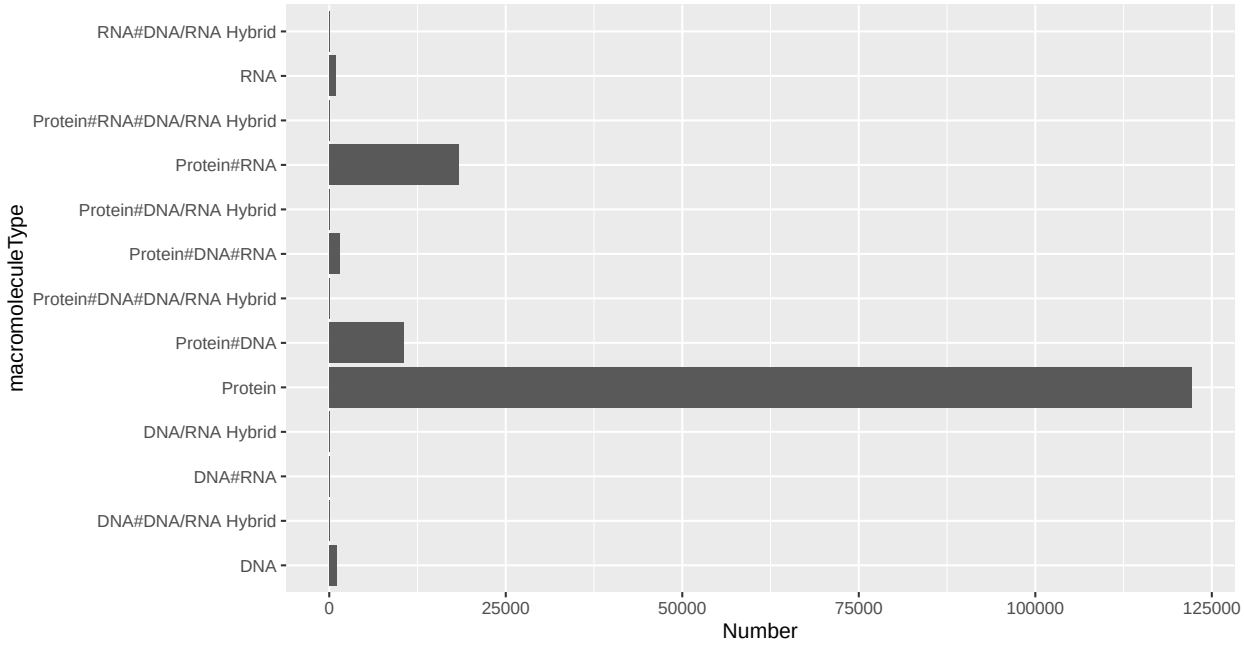
Fig. 3 Distibution of ph in top 15 classifications



2.2.4 Macromolecule

From figure 4, we can see that most of the sample in our dataset is protein along with some DNA and RNA.

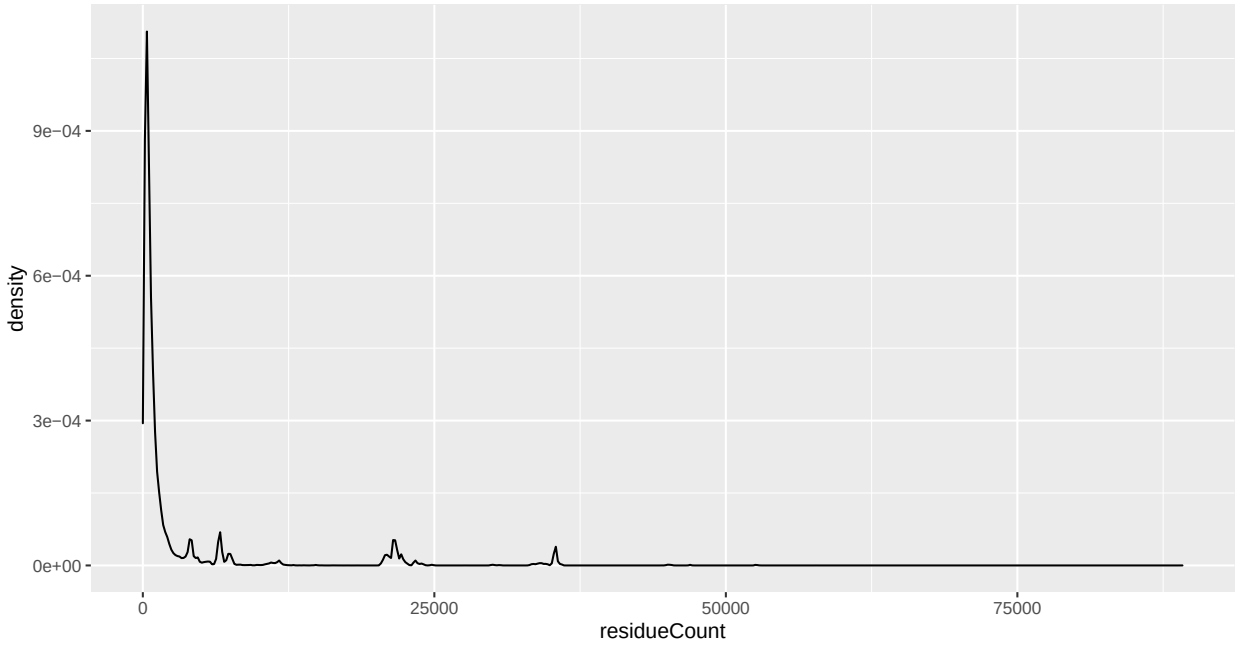
Fig. 4 Number in different macromolecule type



2.2.5 Residue count

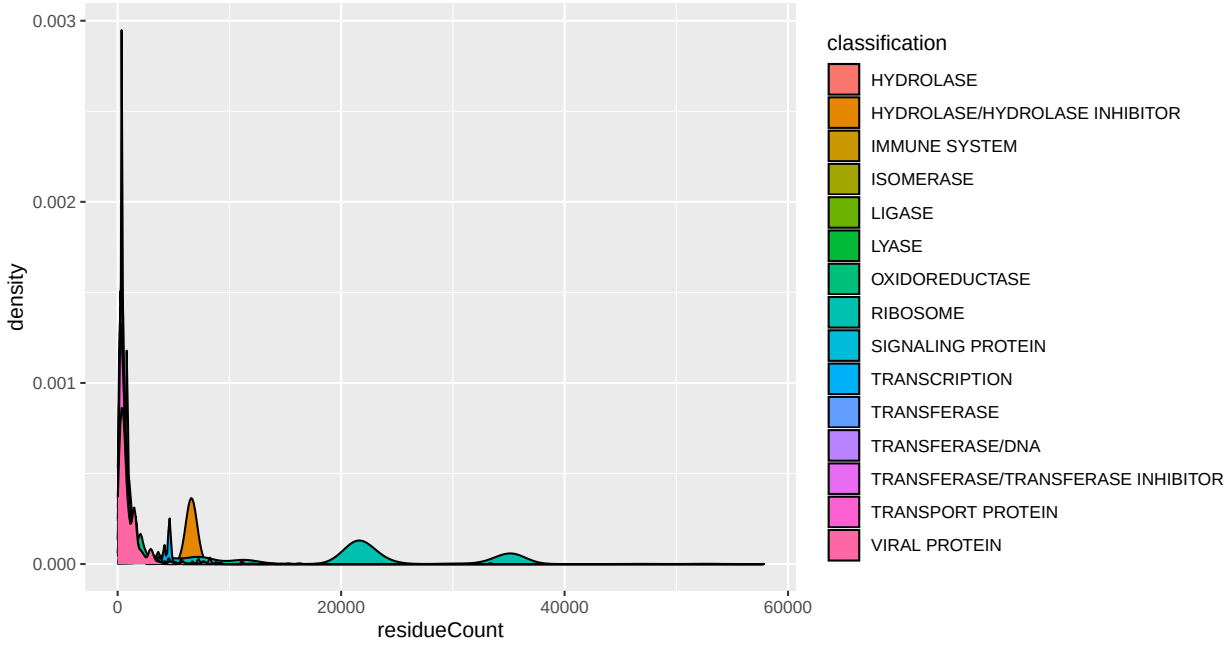
The distribution of residue count in all classification is heavily right skweded as we can see from figure 5. The majority of data is where residue count is lower than 12,500.

Fig. 5 Distribution of residue count



The distribution of residue count in the top 15 classification is right skewed as well. From figure 6, we can see that the outlier in this plot seem to be ribosome class, which has higher residue count from the rest of others classification.

Fig. 6 Distribution 0f residue count for top 15 classifications



2.2.6 Resolution

From our data, the distribution of resolution is right skewed distribution that have median of 2.25. The distribution of resolution in the top 15 classification is similar but the outlier are also Ribosome class like in residue count distribution.

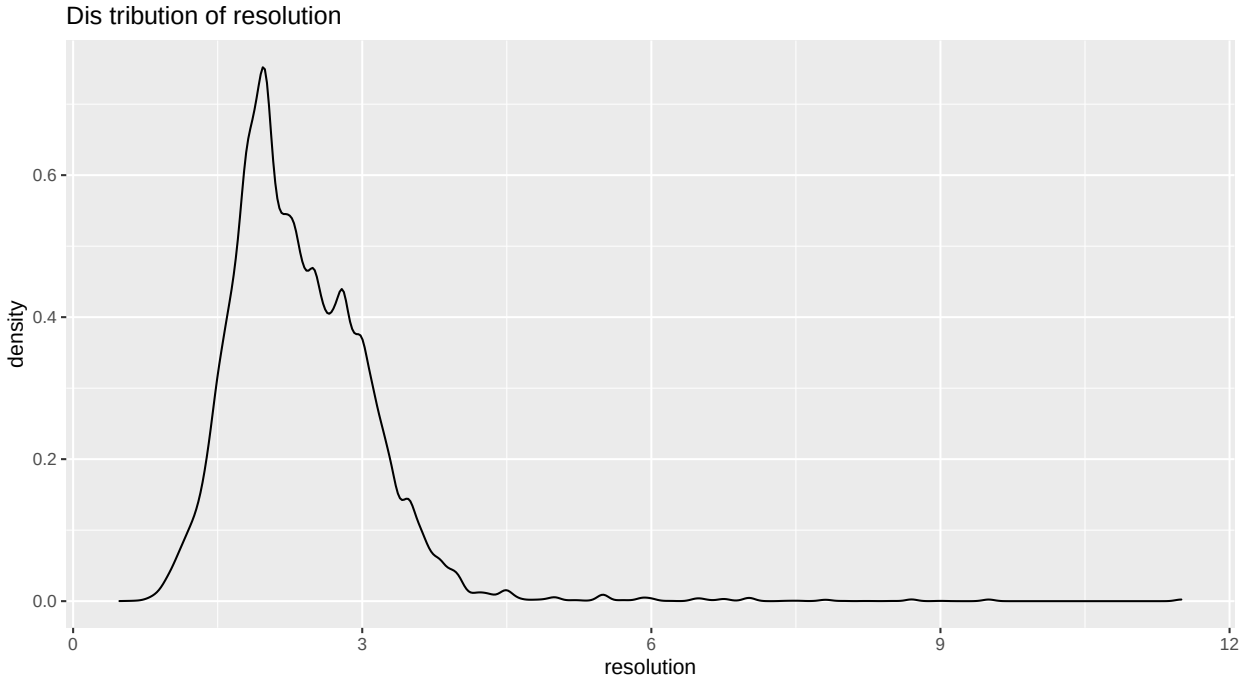
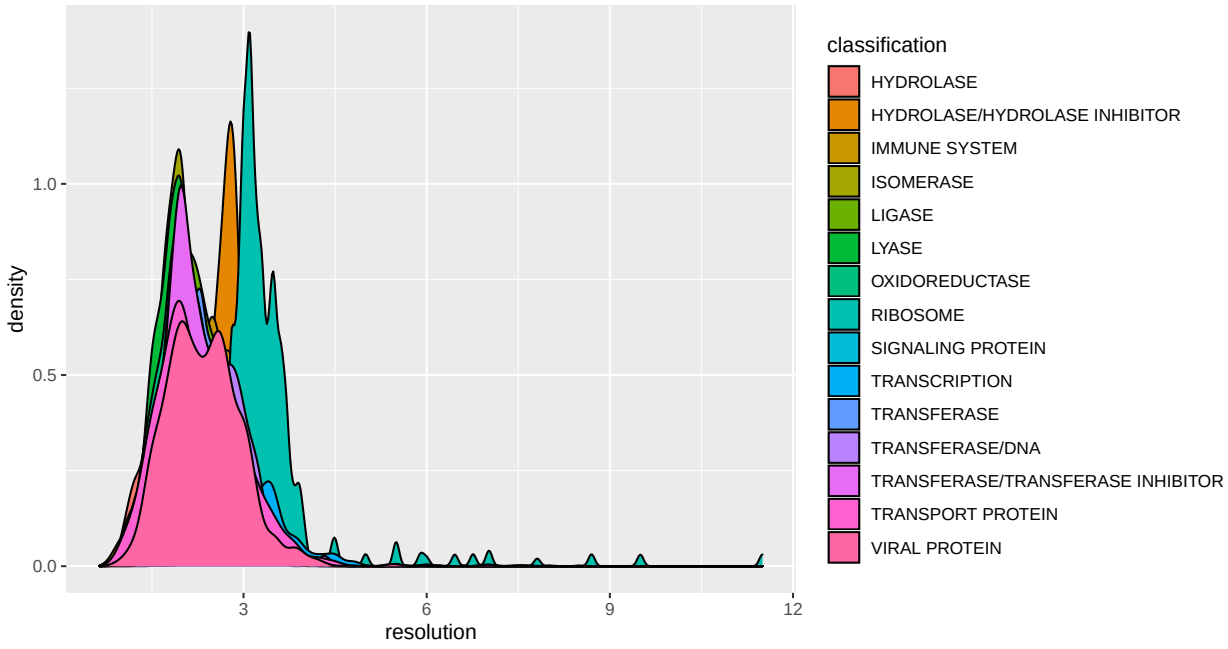


Fig. 7 Distribution of resolution for top 15 classifications



2.2.7 Molecular weight

The distribution of molecular weight in all classification and top 15 classification are similar to each other. The molecular weight is right skewed distribution with ribosome class as outlier as the previous 2 variables as we can see from figure 8. If we look when molecular weight is lesser than 1 millions in figure 9, we can see that there are 3 class that can be seperate from other class, which is Ribosome, Hydrolase/hydrolase inhibitor and transcription.

Fig. 8 Distribution of molecular weight in top 15 classification

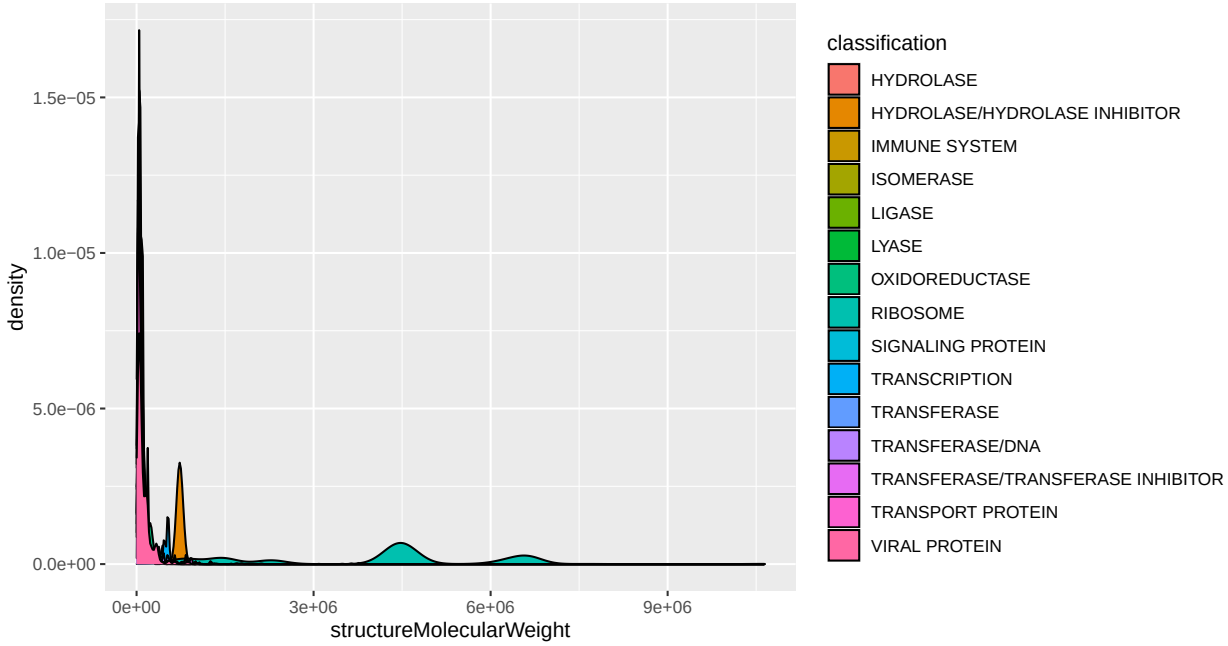
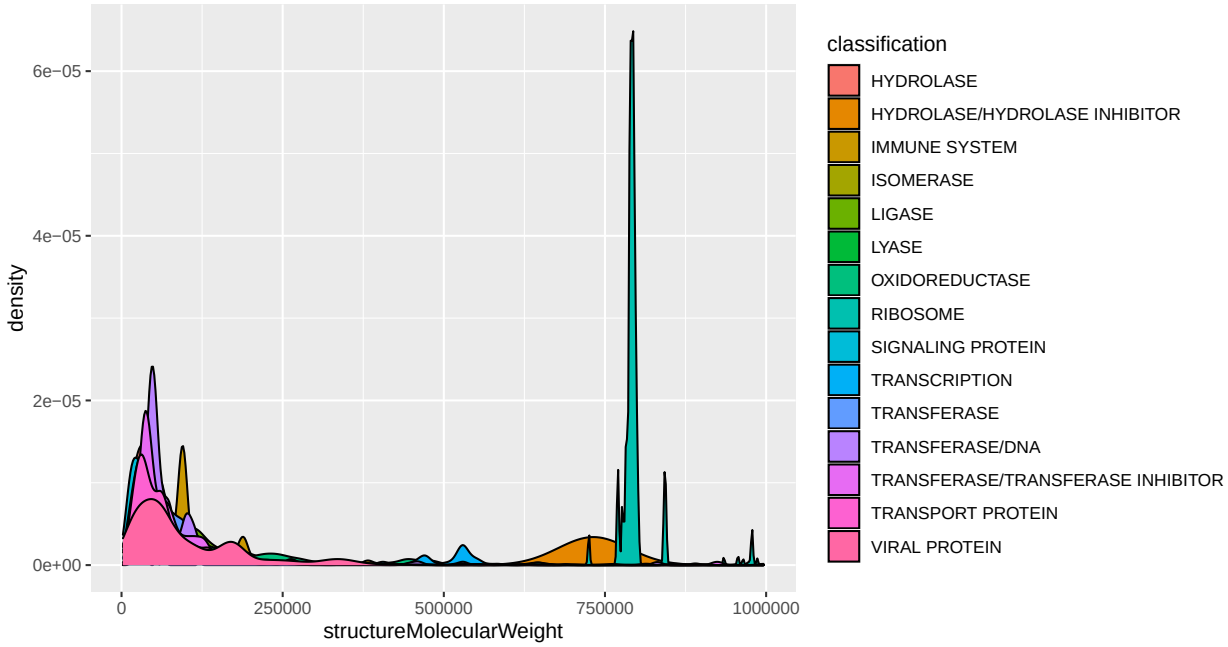


Fig. 9 Distribution of molecular weight under 1 million in top 15 classification



2.2.8 Density Matthew

The distribution of density matthew in all classification and in top 15 classification is approxiamtely normal distributed. As we have describe before, density matthews is relate to type of macrostructure. The majority of our sample have protein macrostructure, so that can explain by why all the classificaion seem to be group up to each other in density matthew distribution.

Fig. 10 Distribution of density matthew

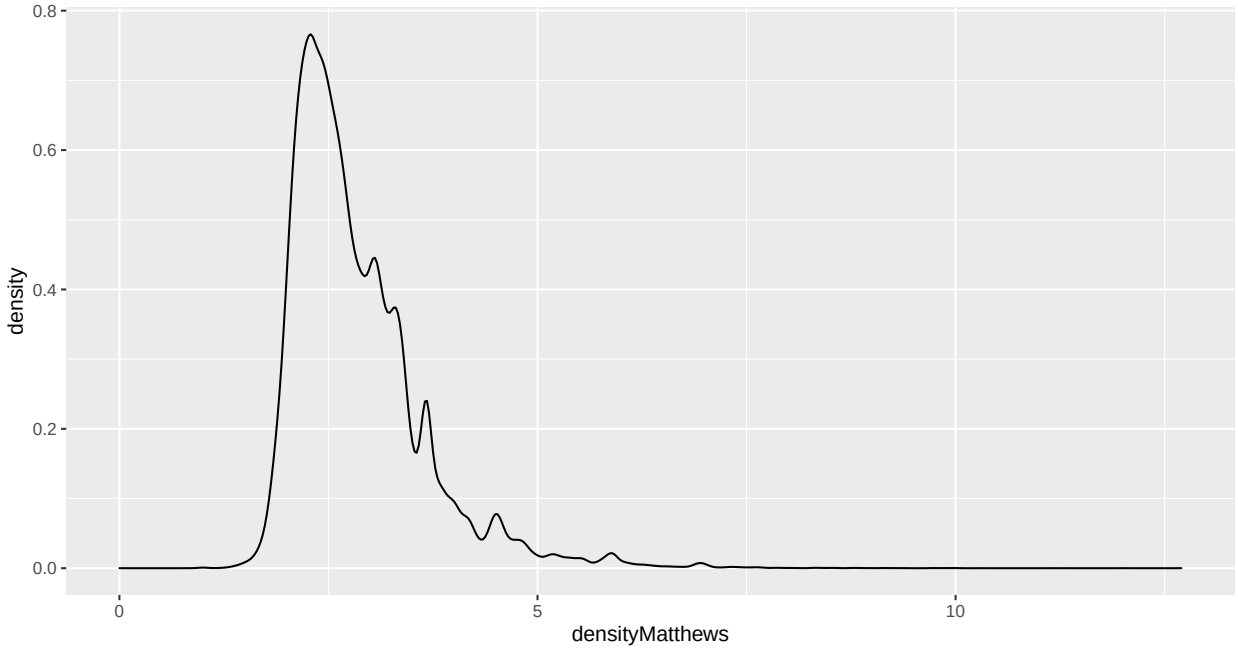
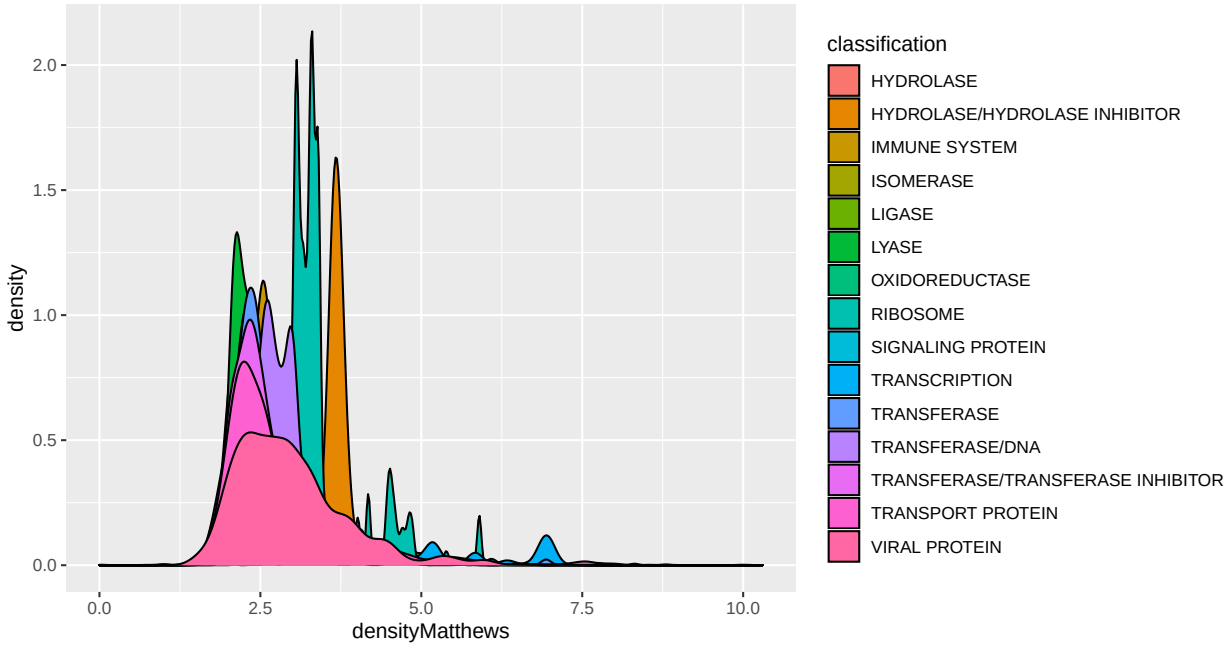


Fig. 11 Distribution of Density matthew of top 15 classification



2.2.9 Density percent sol

The distribution of Density percent sol in all classification and in the top 15 are similar. All the classification are normal distributed and don't have any class that distinguish from the others.

Fig. 12 Distribution of Density percent sol

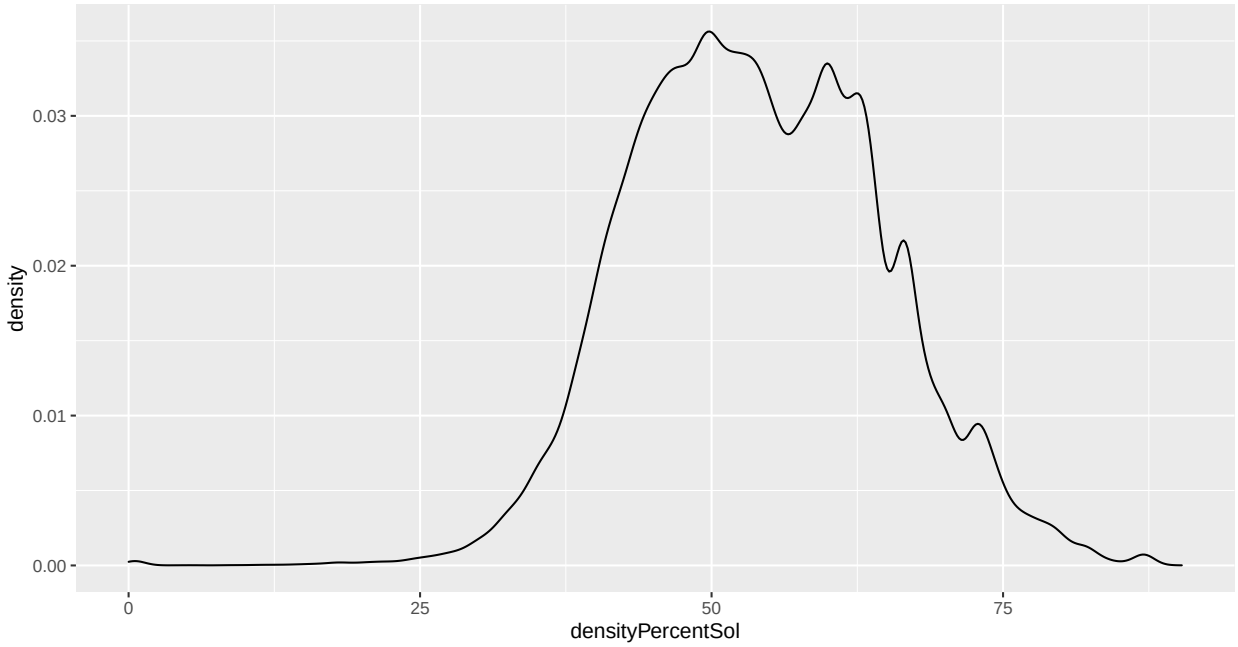
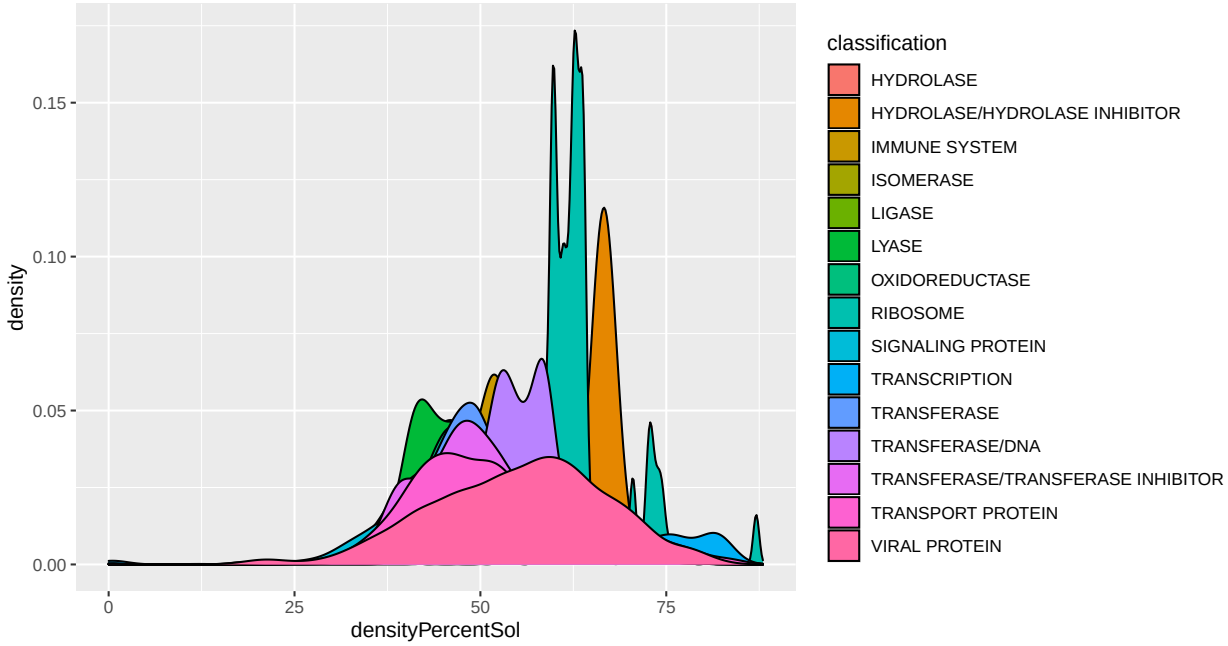


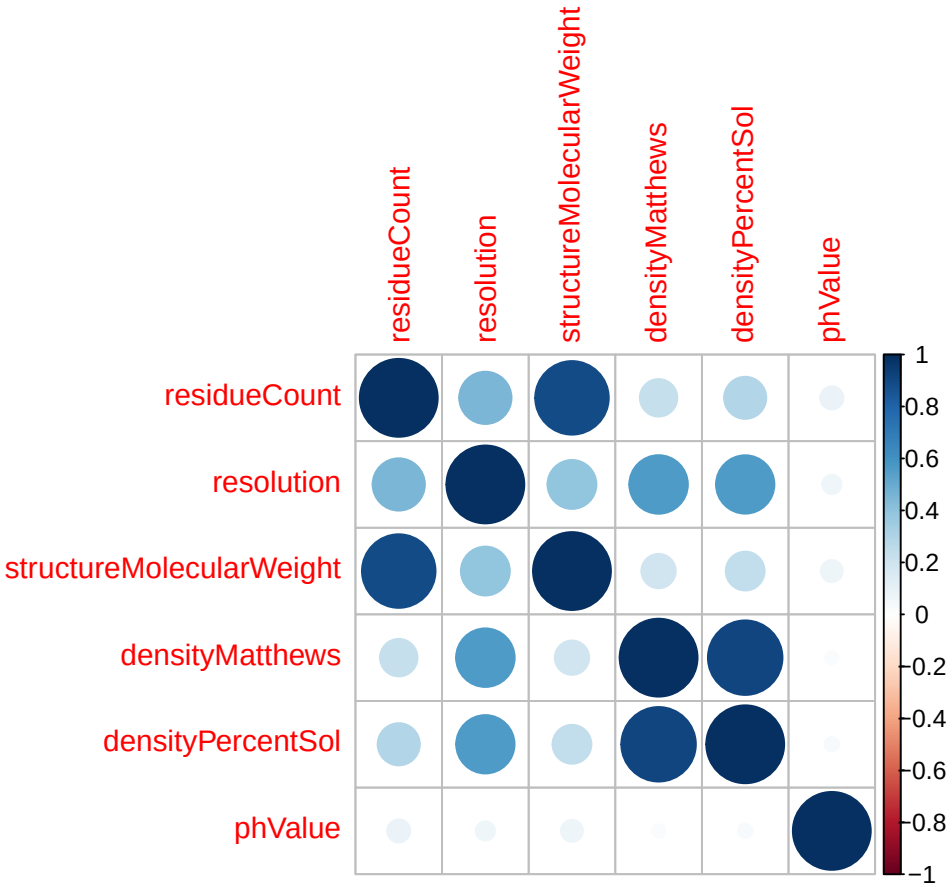
Fig. 13 Distribution of density percent sol in top 15 classifications



2.2.10 Pearson correlation matrix

After we have explore the distribution of all the variables, we will look at the correlation of each variables. From correlation matrix below, we can see that residue count and Molecular weight have a strong correlation with correlation coefficient equal 0.8919. Another pair of variable that have a strong correlation with correlation coefficientequal 0.9156 is density matthews and density percent sol. This two variables are expected to be highly correlate because its both measure the density but use diffrent denominator.

```
##                          residueCount resolution structureMolecularWeight
## residueCount               1.00000000 0.45113474               0.89195777
## resolution                 0.45113474 1.00000000               0.39530766
## structureMolecularWeight   0.89195777 0.39530766               1.00000000
## densityMatthews            0.23270769 0.56590813               0.19527720
## densityPercentSol          0.29325663 0.56199545               0.24966430
## phValue                    0.08849592 0.06168099               0.07985366
##                          densityMatthews densityPercentSol    phValue
## residueCount                  0.23270769         0.2932566 0.08849592
## resolution                    0.56590813         0.5619955 0.06168099
## structureMolecularWeight      0.19527720         0.2496643 0.07985366
## densityMatthews               1.00000000         0.9156367 0.02889776
## densityPercentSol             0.91563665         1.0000000 0.03482220
## phValue                       0.02889776         0.0348222 1.00000000
```



2.3 Model creation

2.3.1 K-Nearest Neighbors(knn)

The concept of K-Nearest Neighbors(knn) is to estimate the conditional probabillity of the prediction by using k nearest points to the point that we want to predict. The value of k that we are going to use is determined by using cross validation.

2.3.2 Decision tree

The concept of decision tree is partition the data into each outcome using diffrent variable to define which sample should be in which catagories. The prediction is based on the most common classification among the training set observations within the partition. To choose that partition, the model will try to minimize two metrics, which are Gini and entropy.Another parameter that use to tune decision tree is complexity parameter(cp). Complexity parameter is defined by a minimum of how much the gini index and entropy must improve for another partition to be added. The formula for Gini index and entropy index are shown below:

$$Gini(j) = \sum_{k=1}^{K} \hat{p}_{j,k}(1 - \hat{p}_{j,k})$$

$$entropy(j) = -\sum_{k=1}^{K} \hat{p}_{j,k} log(\hat{p}_{j,k}) \text{ , with } 0 \times \log(0) \text{ defined as } 0$$

2.3.3 Random forest

Random forest is a machine learning approach that help with the shortcomings of decision tree by averaging multiple decision tree to make the prediction. To induced randomness in each decision tree, the model use bootsrap aggregation. The process to induced randomness consisited with 2 step. The first step is sampling N observations from the training set with repalcement. The second step is randomly select the features to be included into building each decision tree.

3. Model creation and evaluation

To bulid our model, first we will split pdb dataset into train set and test set. The train set accounted for 80 percents of pdb data and the 20 percents are test set. We will will keep only residue count, resolution, molecular weight, density matthew, density percent sol and phvalue for training purpose, other column will be drop. The code for these step are shown below.

```
#we will keep only residue count, resolution, molecular weight,
#density matthew, density percent sol, phvalue for training purpose
temp_1 <- pdb %>%
  filter(classification %in% name_class) %>%
  subset(select=-c(sequence,experimentalTechnique,macromoleculeType))

# test set will be 20% of pdb data
set.seed(1, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(1)` instead
test_index <- createDataPartition(y = temp_1$classification, times = 1, p = 0.2, list = FALSE)
train_set <- temp_1[-test_index,]
temp_2 <- temp_1[test_index,]
```

```r
# Make sure classification in test set are also in train set
test_set <- temp_2 %>%
  semi_join(train_set, by = c("classification"))

# Add rows removed from test set back into train set
removed <- anti_join(temp_2, test_set)
train_set <- rbind(train_set, removed)
```

3.1 K-Nearest Neighbors(knn)

The first model that we are going to create is knn. We going to use cross validation 10 times with 90 percents of data as train part and 10 percents validation part in each time. The best value of K that have the most accuracy will be selected. The code for creating knn model are shown below:

```r
#knn
control_knn <- trainControl(method="cv",number = 10,p=0.9)
train_knn <- train(classification ~ . ,
                   data = train_set,method="knn",
                   trControl=control_knn,
                   tuneGrid=data.frame(k=seq(3,9,2)))
results <- data.frame(method="knn",
                      Accuracy=confusionMatrix(
                        predict(train_knn,test_set),
                          as.factor(test_set$classification))$overall[["Accuracy"]])
results
```

```
##   method  Accuracy
## 1    knn 0.5088698
```

From the results, we get the best accuracy of 0.5069 with k equal 3.

3.2 Decision tree

The second model we are going to create is decision tree. The tuning parameter we are going to tune for decision tree is complexity parameter(cp). As we have describe before cp is defined by a minimum of how much the gini index and entropy must improve for another partition to be added. The code for creating decision tree model are shown below:

```r
train_rpart <- train(classification ~ . ,
                     data = train_set,method="rpart",
                     tuneGrid=data.frame(cp=seq(0.0,0.1,len=25)))
results <- rbind(results,
                 data.frame(method="Decision tree",
                            Accuracy=confusionMatrix(
                              predict(train_rpart,test_set),
                              as.factor(test_set$classification))$overall[["Accuracy"]]))
results
```

```
##          method  Accuracy
## 1           knn 0.5088698
## 2 Decision tree 0.5869450
```

```r
varImp(train_rpart)
```

```
## rpart variable importance
##
##                           Overall
## structureMolecularWeight  100.00
## residueCount               94.69
## densityPercentSol          43.35
## densityMatthews            40.90
## resolution                 20.82
## phValue                     0.00
```

The accuracy that we got from decision tree model is 0.5869, which is better than knn model. The cp that give the best accuracy is 0, which means we will add as much as partition that we could to make the best accuracy. The variable important in the decision tree model are as shown above with molecular weight have the most importance. The following variable that almost got the same variable importance is residue count. The least important variable is pH values.

3.3 Random Forest

The last model that we are going to built is random forest model. This model are created to improve from our previos model, decision tree. For tuning random forest model, we will use cross validation for 5 time to find the best mtry value. Mtry is number of variables randomly sampled as candidates at each split. We are going to create 150 decision trees in this random forest model. The code to build this model are shown below:

```r
#random forest
control_rf <- trainControl(method = "cv",number = 5)
train_rf <- train(classification ~ . ,
                  data = train_set,
                  method="rf",
                  ntree=150,
                  trControl=control_rf)
results <- rbind(results,
                 data.frame(method="Random Forest",
                            Accuracy=confusionMatrix(
                              predict(train_rf,test_set),
                              as.factor(test_set$classification))$overall[["Accuracy"]]))
results
```

```
##          method  Accuracy
## 1           knn 0.5088698
## 2 Decision tree 0.5869450
## 3 Random Forest 0.7331634
```

```r
varImp(train_rf)
```

```
## rf variable importance
##
##                           Overall
## structureMolecularWeight 100.000
```

```
## residueCount              54.038
## densityPercentSol          8.489
## phValue                    2.577
## resolution                 1.192
## densityMatthews            0.000
```

The accuracy from random forest model is 0.7331, which increased from our past two models substaintially. The best mtry for our model is equal to 4. The variable importance that we have in random forest model is similar to decision tree model. The molecular weight is still the most important variable, following by residue count but it has less importance in randoforest model compare to decision tree model. The remaining 4 variable get very low importance in our random forest model.

Looking at confusion matrix of all three of our models, we can see that all three models have high specificity in every classification with more than 90 percents. Ribosome and Transferase/DNA class also have highrer than 90 percents accuracy in all of our model. The ribosome class as we have seen in data visualiztion section, is a class that usally have their data as outlier. This maybe make prediction for ribosome easier than other class.

4. Summary

In this study, We have found that the majority of technique in viewing protein structure is X-ray diffraction. In pdb database, there are over 3,000 classifications, but top 15 classification are accounted for 60 percents of all samples. The top 15 classifications has distribute evenly in pH catagories which crystal was grown. The majority of macromolecule is protein. The distribution of residue count and resolution are right skewed, the outlier are in Ribosome class. The molecular weight distribution in distinguish from other classes in the distribution. The distribution of density matthews and density percent sol are approximately normal distributed. When compute the correlation in each variable we see 2 pairs that have strong correlation. The first pair is residue count and molecular weight with correlation coefficient of 0.891. The second pair is density matthews and density percent sol with correlation coefficient of 0.915. The most variable importance in random forest model is molecular weight, follow by residue count.

In classification model creation, we had built 3 models consisiting with knn, decision tree and random forest. For knn model, we use cross validation to select value of k that had highest accuracy. We got the best accuracy of 0.5069 with k equal to 3. For decision tree model, tuning parameter that were used is complexity parameter. The best cp for our decision tree model is 0, which give 0.5869 overall accuracy. The highest variable importance in decision tree model are molecular weight, follow by residue count. The last model we built is random forest model. The tuning parameter that were used is mtry and we make 150 decision tree. The best accuracy for random forest model is 0.7331 with mtry equal to 4.

4.1. Limitation

The first limitation is we didn't use the sequence in to classifying the class. Although sequence might have a strong correlation with each class, but the data are to big to train on. To incorporate sequence into training will take a lot of computation and run time.
The second limitation is lack of predictor in data. From the data we have only one or two variables that didn't have the distribution of each class overlay on each other. To find more predictor would improve our classification.

4.2 Future work

For future work we can try using sequence as another predictor in our model. But we may have to find a way to clean the sequence data or find model that doesn't take too much time. Other model such as Quadratic

discriminant analysis (QDA) or extra trees classifier could be implemented to see if the accuracy will increase or not. Ensemble multiple model to help us classify could be done to improve the overall accuracy as well.

References

1. Irizarry R. Introduction to data science.

2. Structural Protein Sequences [Internet]. Kaggle.com. 2020 [cited 19 June 2020]. Available from: https://www.kaggle.com/shahir/protein-data-set

3. Bank R. RCSB PDB: Homepage [Internet]. Rcsb.org. 2020 [cited 21 June 2020]. Available from: http://www.rcsb.org/pdb/

4. Residue number - CMBIwiki [Internet]. Www3.cmbi.umcn.nl. 2020 [cited 21 June 2020]. Available from: https://www3.cmbi.umcn.nl/wiki/index.php/Residue_number#:~:text=The%20average%20protein%20contains

5. Resolution - Proteopedia, life in 3D [Internet]. Proteopedia.org. 2020 [cited 21 June 2020]. Available from: https://proteopedia.org/wiki/index.php/Resolution#:~:text=In%20X%2Dray%20crystallography%2C%20resolu

6. Characterizing Your Crystal [Internet]. People.mbi.ucla.edu. 2020 [cited 21 June 2020]. Available from: https://people.mbi.ucla.edu/sawaya/tutorials/Characterize/characterize.html