Ali Ekrem Yesilkanal
Programming Assignment 1 - 10.25.17

Goal: To write a classifier that, given some mutational information about a case, predicts the primary site associated with the cancer, and to evaluate this classifier's performance.
Software: R Studio

STEP-1: Cleaning and exploring the data:
The GDC data provided for this assignment in JSON structure had data from over 9000 cancer patients across 26 distinct cancer types (described as the "primary site" of the tumor). Each patient has a list of mutations, and the locations of the mutations on the genome as Ill as the genes they are associated with are given in the data set. I need to build a classifier that takes features measured from the given mutational data, and accurately predict the primary site of a patient's tumor. The 26 primary sites present in the data set constitute the classes I will be using as the dependent variable as I train our classifier.
The JSON data structure has lists within lists , so the first thing I need to do is to flatten the data set and extract the necessary information into a data frame in R (Table 1).

mut_id	case_id	primary_site	gene_symbol	chromosome	mut_position	change
0c21c8d3-2c86-5b1f-911a-1adc5f3037c8	0878e44d-bdc8-4d5f-9bf4-31101f14f797	Uterus	EFR3A	chr8	131940576	G>T
f1d8c1d1-09a4-515c-a551-d93dd0fe7c22	166e76db-ccd8-4760-a517-d2bc8937ea29	Brain	UBAP2	chr9	33986775	G>A
e1ad55e3-0607-5333-9f0f-2d40c3753da6	ded3feb2-1079-4520-a7ca-f5b5fc73d7c5	Colorectal	CPS1	chr2	210605162	G>T
0c340f12-d9e7-5fa9-84bc-7d84570b984a	b045f24b-f822-4df9-9ffa-47308edcec8c	Uterus	CBWD1	chr9	162470	C>A
e4dc1a8e-5db5-5ad7-b36e-2be04c11bcb5	0712566e-3371-4715-95f1-5b792e72d758	Colorectal	GALNT13	chr2	154245942	G>T
71da99d6-db70-51b2-a6a7-2b3f484437d2	eb0eb159-732e-43ff-a402-7f2d9f67daa8	Cervix	GSTA4	chr6	52987399	C>A
c16fad98-5412-5f97-95f2-ef0236d6c181	02b7b0c5-dad6-4270-b56b-3d04285e8147	Soft Tissue	MTUS2	chr13	29033938	T>A

….
Table 1: Flattened and cleaned data.

Each row of this data frame has information on a single specific mutation, and all the mutations are indexed by individual "id"s (column 1). Patients that these mutations belong to are indexed with a "case_id" (column 2). Each patient has multiple genes mutated and some genes are mutated more than once (Tables 2 and 3). Since these mutations are single nucleotide mutations, I reduced mutation "start" and "end" position to a single location column called "mut_position".

case_id	primary_site	gene_symbol	chromosome	mut_position	change
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	LVRN	chr5	115983286	G>T
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	IGKV1D-33	chr2	89914358	T>A
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	ARVCF	chr22	19973170	G>T
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	FAM111A	chr11	59152023	G>T
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	NOTUM	chr17	81957019	T>A
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	ZMIZ1	chr10	79305232	G>C
0004d251-3f70-4395-b175-c94c2f5b1b81	Liver	THBS2	chr6	169220339	T>G
000d566c-96c7-4f1c-b36e-fa2222467983	Prostate	GATC	chr12	120459958	G>C
0011a67b-1ba9-4a32-a6b8-7850759a38cf	Colorectal	APC	chr5	112839627	G>T
0011a67b-1ba9-4a32-a6b8-7850759a38cf	Colorectal	GPR157	chr1	9105558	C>T
0011a67b-1ba9-4a32-a6b8-7850759a38cf	Colorectal	ZNF493	chr19	21423744	C>G

….
Table 2: Mutations grouped by the case id.

case_id	primary_site	gene_symbol	chromosome	mut_position	change
0011a67b-1ba9-4a32-a6b8-7850759a38cf	Colorectal	ZNF714	chr19	21117893	C>G
0011a67b-1ba9-4a32-a6b8-7850759a38cf	Colorectal	ZNF714	chr19	21117148	C>T
0011a67b-1ba9-4a32-a6b8-7850759a38cf	Colorectal	ZNF714	chr19	21117725	C>G

Table 3: An example of the same gene being mutated at multiple positions within the same case id.

STEP-2: Building features for the classifier
Feature set 1: The type of nucleotide change that takes place during a mutational event can depend on the tissue origin of the cancer. For example UV-B can cause C>T mutations in skin cancers, whereas smoking more frequently causes G>C and T>A changes in lung cancer. Therefore, I first wanted to test if the type of the nucleotide change can classify the cases accurately.

In order to do this, I first counted the frequency of the 9 possible nucleotide changes that can take place for each case_id. Then I selected the highest frequency nucleotide change within a case id as the feature for that

case, and created a boolean feature matrix containing 9 columns for each change. In this matrix the highest frequency mutational change takes a “TRUE” and the rest of the changes take “FALSE”.

After creating the feature matrix, I split the cases (about 9000 cases ids) into training and validation (test) set at the ratio of 0.8:0.2 respectively at random (using *sample()* function in R). Figure 1 demonstrates that each class is represented at similar percentages between the training and the test set. However, within each set “Lung”, “Brain”, “Breast”, “Colorectal” and “Kidney” cases constituted the majority.

```
> table(case.train$primary_site)/nrow(case.train)*100

Adrenal Gland     Bile Duct       Bladder   Bone Marrow         Brain        Breast        Cervix    Colorectal     Esophagus           Eye Head and Neck        Kidney         Liver
    1.6890964     0.4943697     4.4905246     0.9338094     8.4042845     9.9011261     3.0623455     6.0834935     1.9088163     0.5630321     5.5341939     6.9074430     4.0236199
         Lung   Lymph Nodes         Ovary      Pancreas        Pleura      Prostate          Skin   Soft Tissue       Stomach       Testis         Thymus       Thyroid        Uterus
   11.4940950     0.4119747     4.5317221     1.7852238     0.8514144     4.1197473     5.2458116     2.2658610     4.7102444     0.9750069     0.7690195     2.4855809     6.3581434
> table(case.test$primary_site)/nrow(case.test)*100

Adrenal Gland     Bile Duct       Bladder   Bone Marrow         Brain        Breast        Cervix    Colorectal     Esophagus           Eye Head and Neck        Kidney         Liver
    1.6528926     0.5509642     4.4077135     1.1570248     9.7520661    10.4132231     3.3057851     5.0688705     2.3691460     0.3856749     5.1239669     7.2176309     3.4710744
         Lung   Lymph Nodes         Ovary      Pancreas        Pleura      Prostate          Skin   Soft Tissue       Stomach       Testis         Thymus       Thyroid        Uterus
   11.0192837     0.3856749     5.1790634     1.2672176     0.7162534     4.5730028     4.1873278     2.4793388     4.4628099     1.1019284     0.8815427     2.8099174     6.0606061
```

Figure 1: Percent representation of each class betIen the training and the validation set.

In order to train my classifier, I used the random forest method. The *randomForest* package in R is very slow when it comes to large matrices because it uses only one CPU at a time. Instead, I used a package called *ranger* to compute my forest because it handles large feature matrices much more efficiently by using all the CPUs available (in my case, a total of 32 CPUs). I trained my classifier for 9 features over 500 trees, which resulted in a prediction error rate of 0.84. The most important variable in this classifier was having “C>A” nucleotide change as the highest frequency mutation (measured by mean decrease Gini coefficient) (Figure 2).

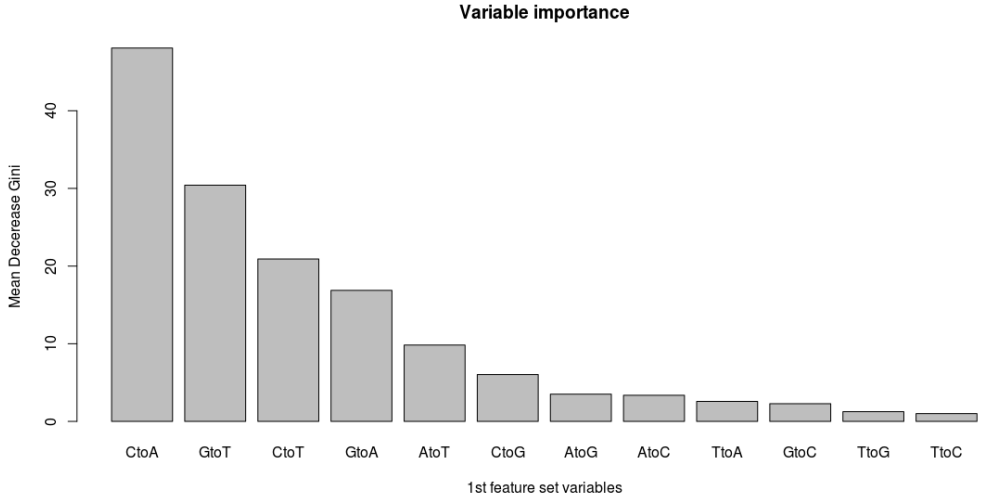

Figure 2: Importance ranking of the variables in the first feature set

The *ranger* package does not provide a function that calculates accuracy. So, I wrote a function that takes percentage of correctly matched cases over total number of cases in a confusion matrix, and called it *accuracy().* When I tested my classifier in the validation set, this first classifier only achieved about 17% accuracy. The confusion matrix showed that almost all of the minor classes were predicted as one of the major classes (Figure 3).

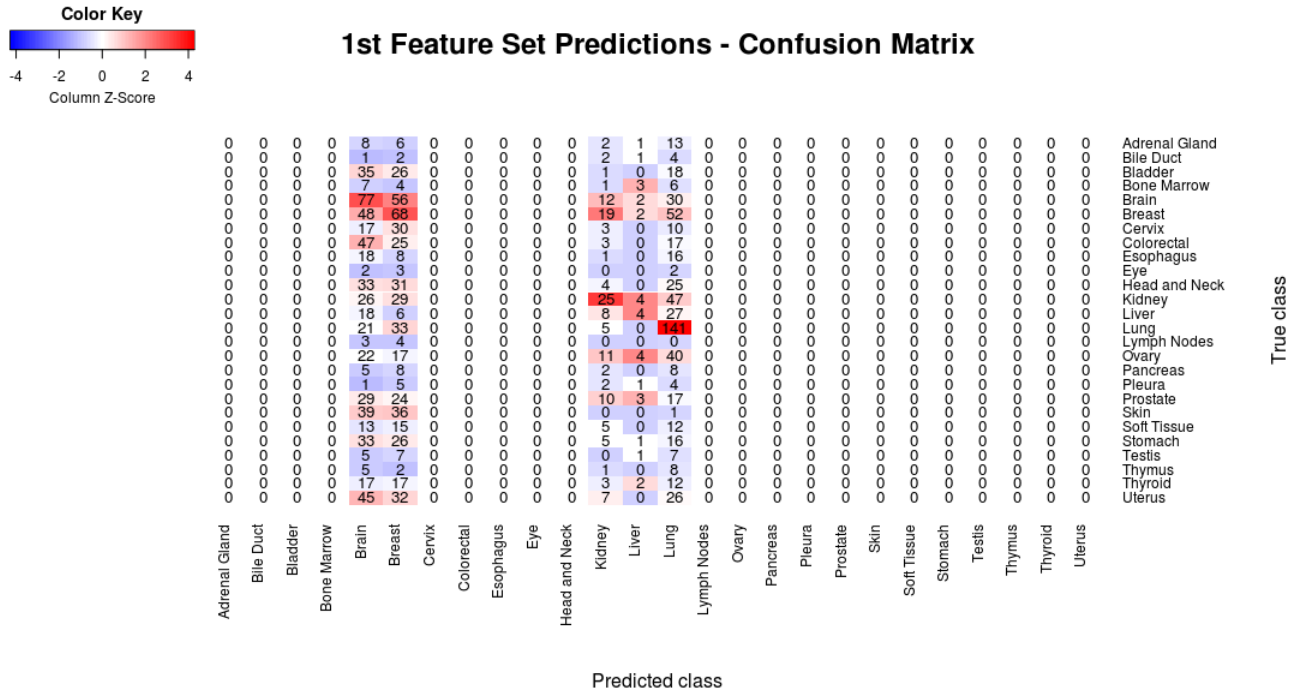

Figure 3: Confusion matrix for the first classifier. Scaled over predicted classes (columns).

Feature Set 2: Since my first classifier performed poorly, I wanted test the genes that are mutated in each case as the features that classify the data. I generated a feature matrix with ~9000 rows (cases) and ~18000 columns (unique gene symbols), where a gene takes the value of "TRUE" if that gene is mutated within a particular case. This analysis was done regardless of the type of mutation or the number of mutations that occurred in a gene.

Training my classifier based on gene features resulted in about 25% accuracy, and the most important variables were genes that are known to be mutated frequently in cancers such as APC, TP53, VHL, and PTEN (Figure 4). Confusion matrix for this classifier shows that more cases within each predicted class were matched correctly (diagonal axis of the matrix) (Figure 5).

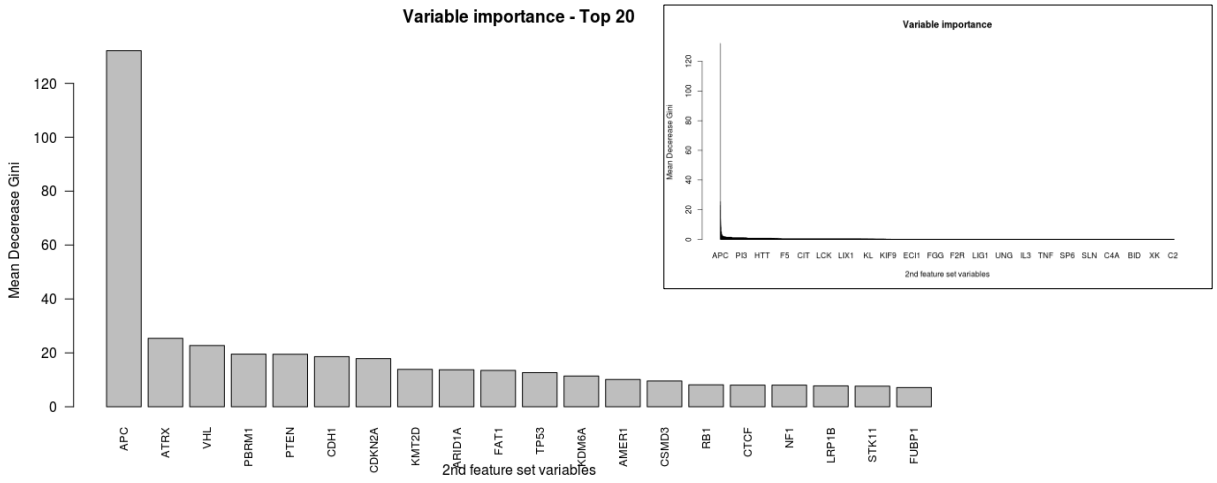

Figure 4: Importance ranking of the variables in the second feature set. Inset: all features (~18,000 genes; bottom panel: top 20 important features.

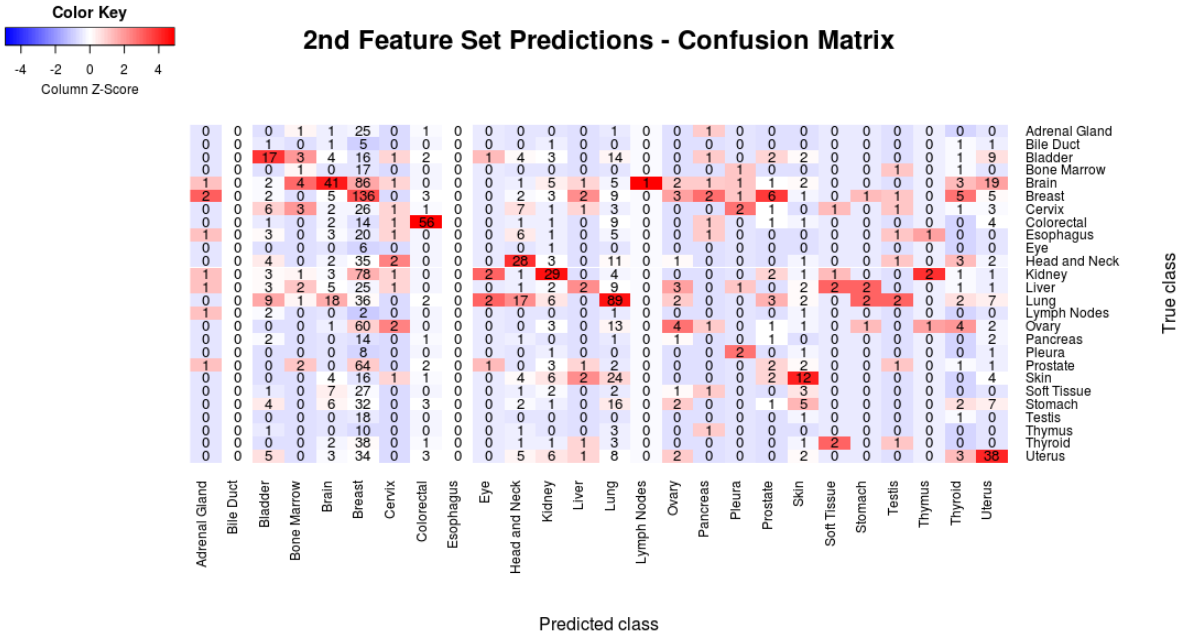

Figure 5: Confusion matrix for the second classifier. Scaled over predicted classes (columns).

9,000 x 18,000 is a large matrix. I wanted to reduce the number of features without affecting accuracy. When I applied a Gini threshold of >1.0 on the gene feature, I was left with 815 genes which still predicted correct classes at the rate of ~25% with a similar confusion matrix (Figure 6).

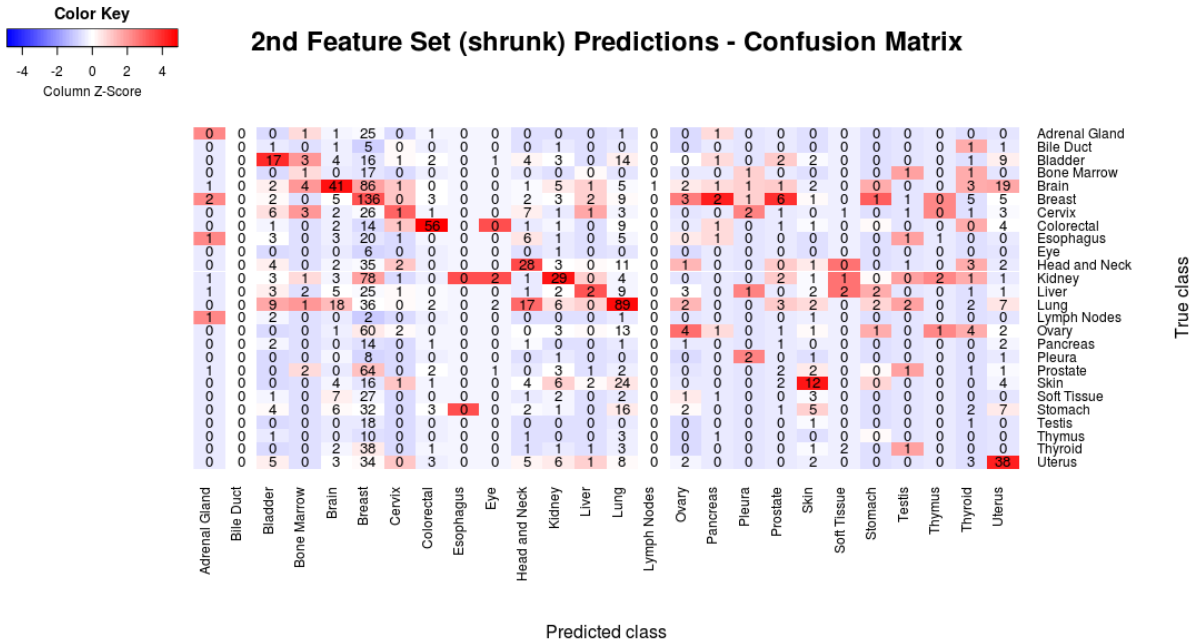

Figure 6: Confusion matrix for the second classifier with only 815 features.
Scaled over predicted classes (columns).

Feature set 3: Certain cancer types can have higher mutational rates. So I added a feature column to the 2nd feature matrix indicating total number of mutations per case id. Adding this column to the feature matrix increased the accuracy of the classifier from 25% to ~28% (Figure 7). Interestingly, number of mutations was much more important for this classifier than the gene features (Figure 8).

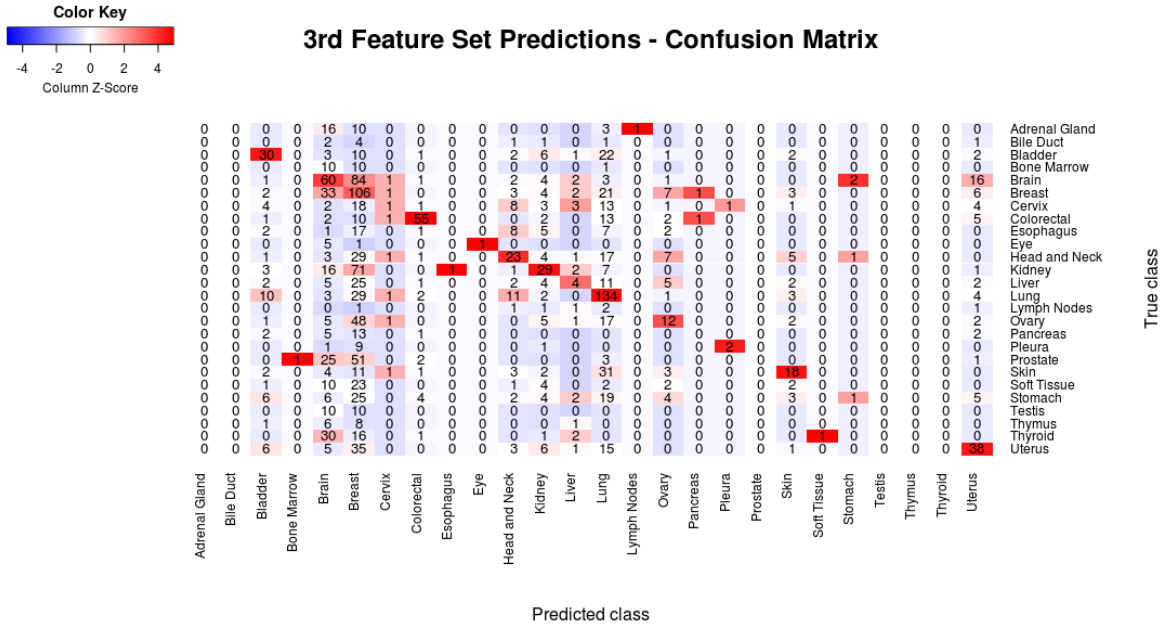

Figure 7: Confusion matrix for the third classifier with 815 gene features and "number of mutations" feature. Scaled over predicted classes (columns).

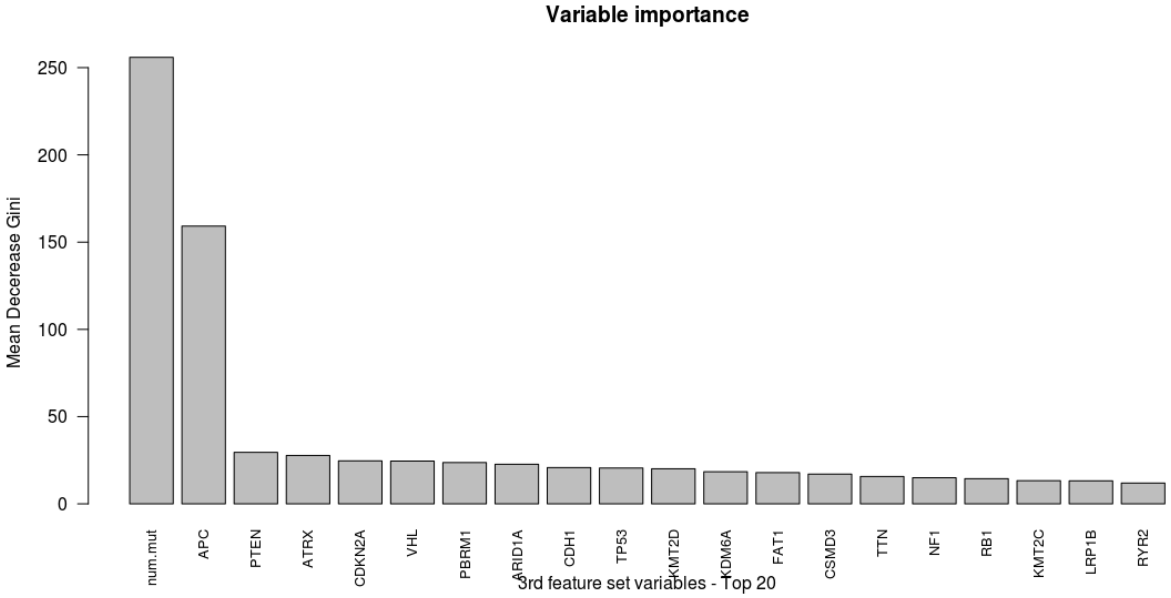

Figure 8: Importance ranking of the variables in the third feature set, demonstrating that "num.mut" feature is much more important than the gene features.

Feature set 4: Combining feature matrices 1 (highest frequency nucleotide change) and 3 (genes and the total number of mutations) column-wise resulted in 30% accuracy (Figure 9). Surprisingly, nucleotide change were more important features than almost all of the mutated genes (except for APC). (Figure 10).

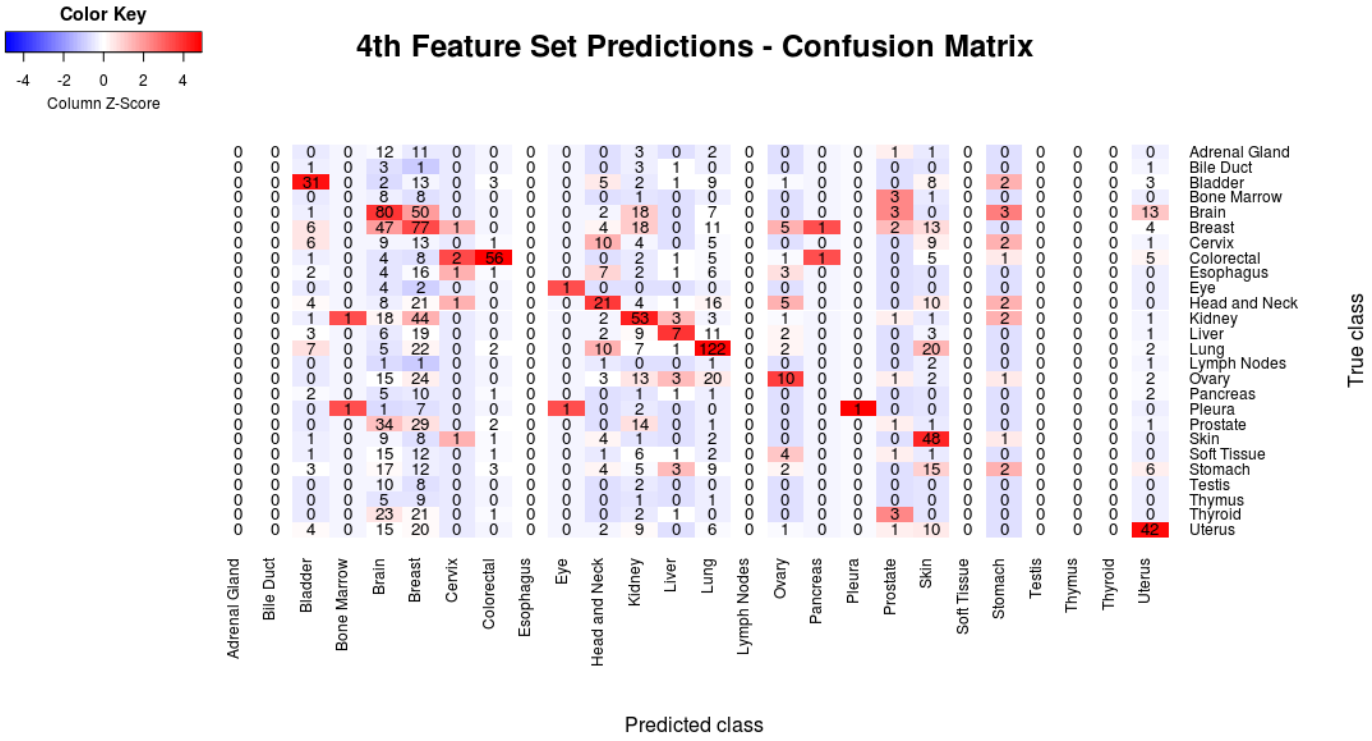

Figure 9: Confusion matrix for the fourth classifier with highest frequency nucleotide change features, gene features, and total number of mutations feature. Scaled over predicted classes (columns).

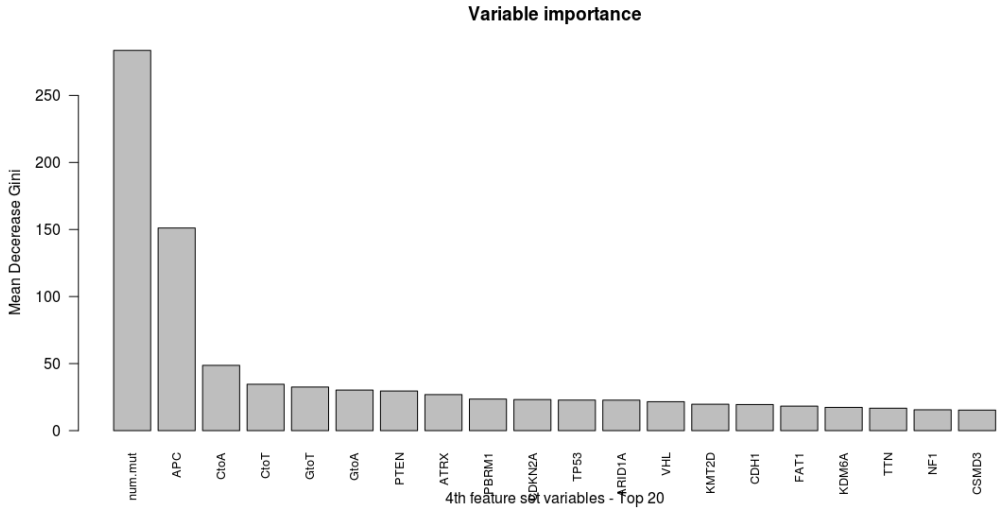

Figure 10: Importance ranking of the variables in the fourth feature set

Feature set 5 and 6: I was curious to see if the chromosomes where the observed mutations reside could classify the primary tissues . By themselves, the 24 chromosome features were predictive of the primary tissue at only 17.8% accuracy. Adding the chromosome features to the feature matrix 4 (*Feature set 6*) did not improve the accuracy of the classifier. However, chromosomes were more important variables than most of the mutated genes in the 6th classifier (Figure 11).

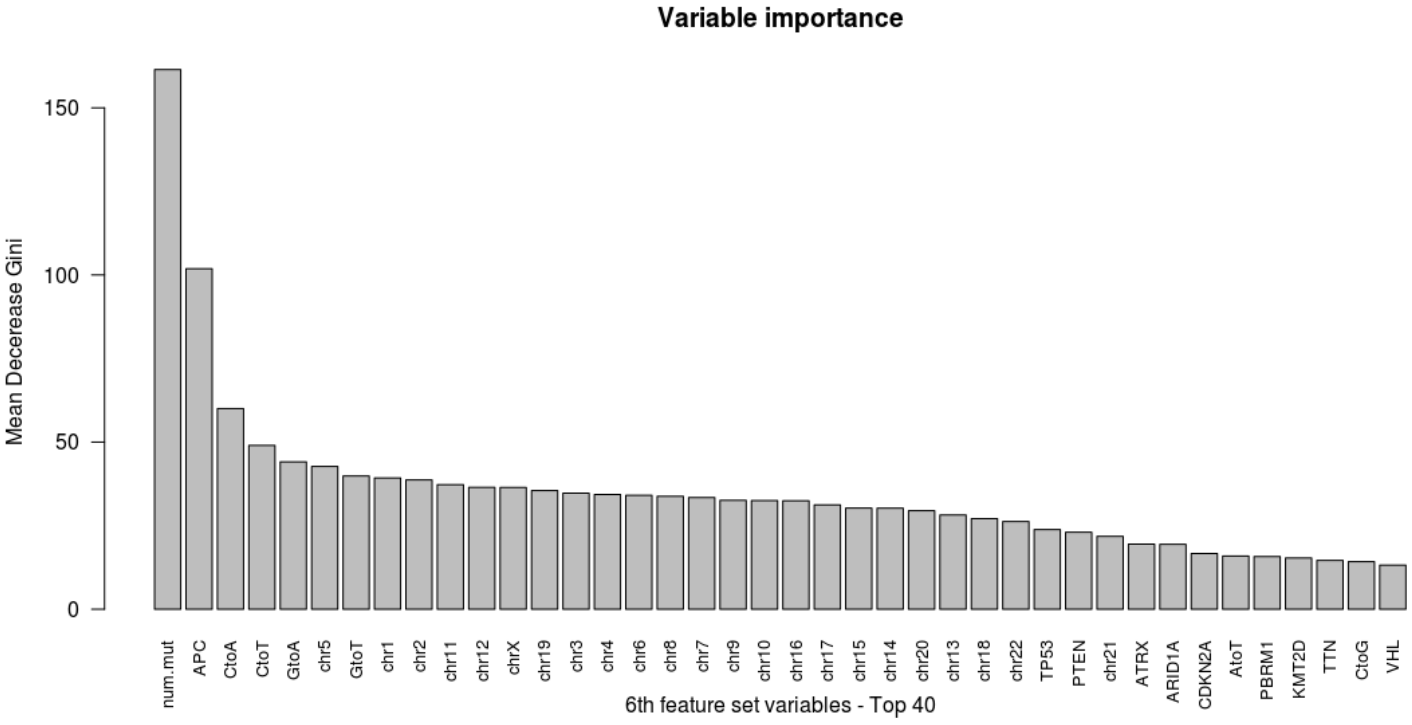

Figure 11: Importance ranking of the variables in the 6th feature set

Conclusions and Discussion

All of the feature matrices I tried to train a classifier that can predict primary site of a tumor based on SNVs have yielded 30% accuracy or less (see summary table 1). Nevertheless, all of these classifiers performed better than random chance (1/26 classes * 100 = 3.8%) Features like total number of mutations, type of the nucleotide change, and chromosomes were better features than mutated genes in predicting primary site of a tumor.

One possible reason for the low accuracy of my classifier is the fact that the data set we are given is unbalanced. Some classes (brain, breast, kidney etc..) have more than 1,000 cases whereas some classes had less than 50. Combining "Lymph Nodes", "Bile Duct", "Eye", "Thymus", "Pleura", "Bone Marrow", and "Testis" classes into one class (called “other”) while training the forest did not improve the accuracy of the classifier (29.4% accuracy). Increasing the number of trees to 1,000 did not have any effect either. Even my best-performing classifiers matched majority of the test cases into the 4 major classes (brain, breast, kidney, lung), as depicted by the row-scaling of the confusion matrix (Figure 12).

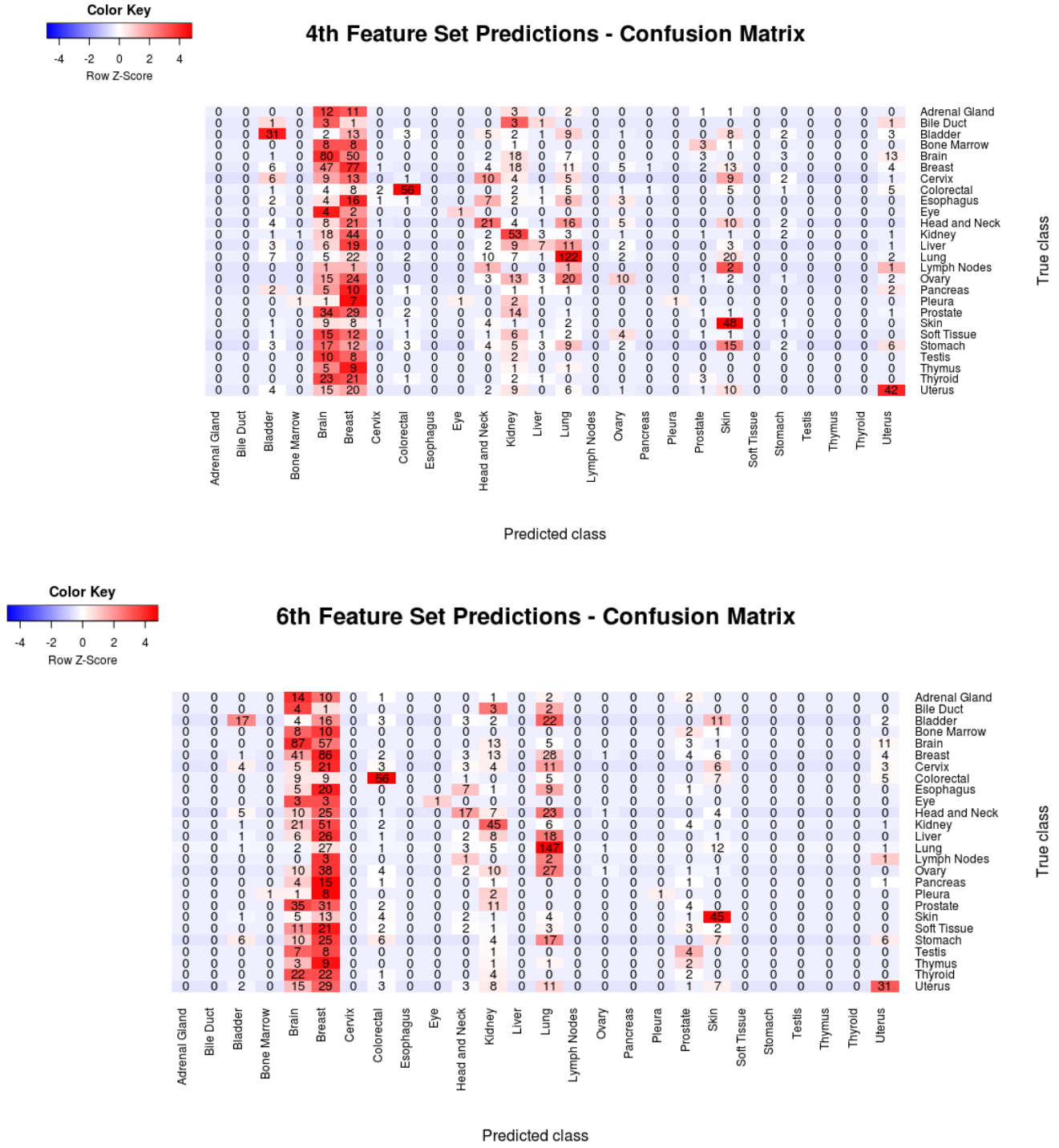

Figure 12: Confusion matrix for the classifiers 4 and 6 (~30% accuracy), scaled over true classes (rows).

To improve accuracy of this classifier, the following approaches can be pursued:

- Under-sampling or over-sampling methods can be incorporated to create more equally weighted classes (particularly by reducing the sample size of lung, breast, brain, and kidney cases)
- RNA expression-based features can be used to predict the primary site of these tumors.

Classifier	Features included	Percent accuracy
Feature set 1	Most frequent nucleotide change	~17%
Feature set 2	Mutated genes	~25%
Feature set 3	Mutated genes + total number of mutations	~28%
Feature set 4	Most frequent nucleotide change + mutated genes + total number of mutations	~30%
Feature set 5	Chromosomes	~17%
Feature set 6	Most frequent nucleotide change + mutated genes + total number of mutations + chromosomes	~30%

Summary figure 1: Feature matrices used to train the classifier, and the corresponding accuracies measured on the validation set.