Group C Report

Alexzandrei Rosario	(Silhouette calculation, k-means, rand index, heatmaps)
Andy Lai	(4 SVM models and analysis)
Dylan Frederick	(KNN models, discussion)
Jeff Maloney	(Lasso, random forest, ensemble and test on dataset 2)
Logan Martinson	(Writing abstract and motivation, proofreading)

~~~~~~

Abstract

Motivation:

In this paper, gene expression data will be analyzed in order to develop a model to predict AML cancer vs non-AML status of a patient. 7 different machine learning models were used for this purpose. Other groups worked on the same problem, providing data useful for feature selection.

Results:

All models tested revealed high consistency in recognizing AML. The most accurate model was Lasso, which achieved an accuracy of 98.2%, while the lowest was SVM Polynomial at 92.7%. Applying machine learning methods to gene expression data appears to be a promising method of diagnosis for AML.

~~~~~~

Introduction

Acute Myeloid Leukemia (AML) is an extremely dangerous form of cancer. Like many diseases, early diagnosis is valuable for effective treatment. Fortunately, many forms of leukemia affect RNA expression in very clear ways, meaning that it should be possible to predict AML, and tell it apart from other cancers, through Gene Expression Profiling (GEP). Many other forms of diagnosis, while reliable, have heavy requirements of manpower or resources, meaning they aren't always scaleable. GEP, meanwhile, requires only a simple blood test. If the results of GEP can be used to consistently identify AML compared to other cancers, it would be invaluable for

the diagnosis of patients on a large scale. Even if not a full replacement for professional human expertise, this could serve to take some pressure off of often overworked doctors. Devising a model for telling AML from non-AML patients is the purpose of this paper.

Methods

The methods we performed for this project utilized various predictive models. We used binary classification of AML vs non-AML, for each patient in the phenotype data matrix we gave the label '1' when column p.info$Disease == 'AML' and '0' otherwise. To build these predictive models, we used an R kernel on Jupyter notebooks. Our first step was to split the dataset into five different folds. We randomly split the patient indexes into five groups. Each fold uses 4 of the groups as training and 1 for testing. We looked at the overall balance of AML vs non-AML in each fold compared to the overall dataset. For each individual fold, we performed feature selection on the training data with different methods. The first feature selection method was with the correlation of the data. For every gene, we correlated the gene expression values for every patient with their class labels. So a list of real numbers is being correlated with a list of 0's and 1's. The second feature selection method is doing a t-test with the data and finding the lowest p-values. For each gene, the gene expression values of AML patients are compared to the gene expression values of non-AML patients. We looked at the Bonferroni adjusted p values and ended up selecting the top 100 genes to use as features for most models. We also compared the lists of genes obtained from different feature selection methods in the different folds using the R method intersect(). Lastly, for each fold, and for feature selection in the fold, we built a predictive model that would predict Acute Myeloid Leukemia, and would output the accuracy of the model. The models we used were Lasso, KNN, Random Forest, and various SVM models.

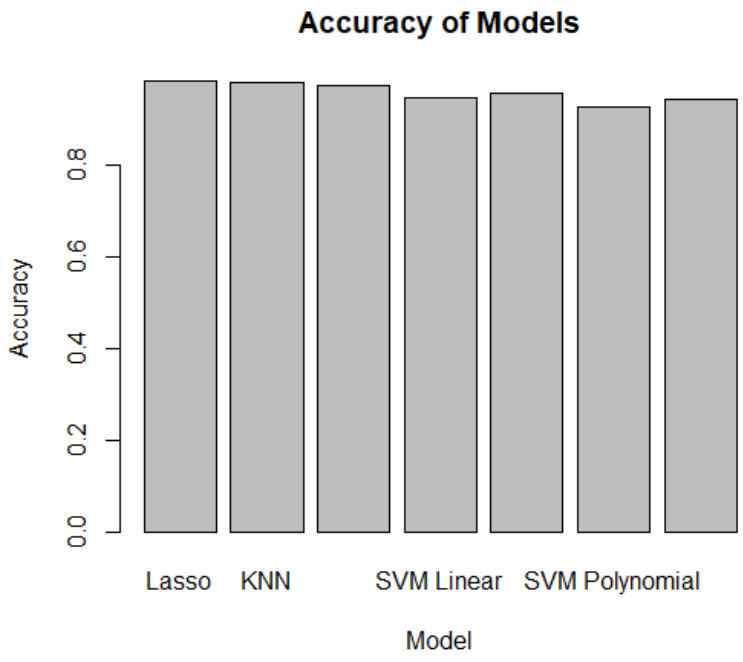
We made two ensemble models. The 'ensemble' was just a method that did a majority vote of each method's prediction vectors. For each model tested in dataset 2, we did not train the model on all of the data from dataset 1, only the training data from the fold it came from. We look at the error rates of each model individually and as an ensemble, in dataset 1 and when applied to dataset 2. The KNN models were built and ran by Dylan Frederick. Jeff Maloney did lasso, random forest, ensemble and test on dataset 2. Andy Lai did 4 types of SVM per fold and analysis.

After all the models described above were used, Alexz took the top 100 genes from the 5th fold from feature selection through the t-test for clustering of the patients. Using the Silhouette Method, we found the optimum numbers of clusters to be 2. This falls in line with our preconceived notion of splitting the patients into AML vs. NON-AML clusters. After this, we clustered the data based on both euclidean distance and correlation between patients. Finally, taking the Rand Index of the two different methods of clustering used. In our case, we got a Rand Index of 0.6533 which indicates the two methods are more similar than not.

In this report, we used SVMs to classify data based on gene expression levels. Specifically, we selected the top 100 genes in each fold using k = 5 cross-validation to train our SVMs. In our case, we used k = 5 cross-validation, which means that we divided our data into five equal subsets and trained our SVMs on four of the subsets, while testing on the remaining subset. By selecting the top 100 genes in each fold, we aimed to identify the genes that were most important for our classification task. We then trained our SVMs using these genes as features and tested their performance on the testing subset. We evaluated the performance of our SVMs for accuracy.

Results

	accuracy
Lasso	0.982
KNN	0.98
Random Forest	0.972
SVM linear	0.945
SVM radial	0.957
SVM polynomial	0.927



SVM sigmoid	0.943

Genome Seeker Results

```
LogisticRegression: 98.56%
RandomForestClassifier: 96.48%
KNeighborsClassifier: 96.16%
XGBClassifier: 97.44%
VotingClassifier: 98.08%
```

Discussion

The error rate of models trained and tested in dataset 1 is quite low. The models tested in dataset 2 did not cross over as well. Random Forest has 8% error in dataset 1, and 24% in dataset 2. SVM linear goes from 5% to 28%. The star performer of generalizing to a new dataset is SVM radial. It went from an error rate of 3.8% to 6.6%. So the error rates increased, respectively, 3x, over 5x, and less than 2x.

Ensemble A			
	Dataset 1 error rate	Dataset 2 error rate	crossover change for the worse
Random Forest	0.082	0.24	2.9x
SVM radial	0.0381	0.066	1.7x
SVM linear	0.0512	0.28	5.5x
ensemble	0.03	0.07	2.3x

In dataset 1 the ensemble had a lower error rate than the individual models. When crossed over to dataset 2, 2 of the models did so poorly that the ensemble performs worse than the best model alone. Perhaps weighted votes for the ensemble would have been better than a majority vote, since one model was so much better than the others.

Another result that stood out was that of two Lasso models, one with all 12708 genes as features compared to one with correlation-selected 33 genes. The model with only 33 genes had an error rate 3x the non-feature selected model for error rates of 0.018 and 0.056.

The overall proportion of AML cases in dataset 1 is 0.39 and the proportion in each k fold is very similar. So each fold was balanced similarly to the overall dataset. An ensemble is tested on dataset 2. The proportion of AML cases here is 0.031, lower than in our training set.

Feature Selection, comparison across folds, and within a fold		
t-test	number of shared genes	
fold 1 vs fold 2	92 / 100	.92
fold1 vs fold 3	93 / 100	.93
fold 2 vs fold 3	93 / 100	.93
correlation		
fold 1 vs fold 3	32 / 33	.97
correlation vs t-test fold 1		
top 37 t-test vs corr	30 / 37	.81
top 96 t-test vs corr	75 / 96	.78

The genes selected in each fold were very similar. For the t-test selection, folds 1 and 2 shared 93 genes, folds 1 and 3, and folds 2 and 3 each shared 92 genes (out of 100). For the correlation selection, folds 1 and 3 we used 33 and 37 genes. 32 of the 33 genes from fold 1 were also in fold 3's list. There is a little more variation selected between the methods. Of the top 37 genes selected by correlation vs top 37 t-test selected genes, 30 / 37 are the same. 75 out of the top 96 genes are the same across the two. More variety of genes are selected by different methods than by different folds, which makes sense. So the K fold feature selection isn't that interesting, but does show that the folds are well balanced.

With our experiment, the Lasso model had the highest accuracy in predicting Acute Myeloid Leukemia. KNN was close with the second highest, and Random Forest being third. All four of the SVM models were noticeably behind the three models previously mentioned, though still above 90%.

When comparing the results of our group's models to the group Genome Seeker's results, our accuracies had some slight differences. For the models that both groups implemented, our group's KNN model was slightly higher, and our Random Forest model was noticeably higher than Genome Seeker. This could be due to different factors, such as the splits of training and testing data, or feature selection thresholds.

When comparing our results with the RNA-seq group using SVM, they achieved an accuracy of 0.93, while our average accuracy is 0.94299. The difference is small, but it should be noted that they used 10-fold cross-validation (k = 10), whereas we used 5-fold cross-validation (k = 5). This difference in methodology might slightly affect the accuracy. Nevertheless, when looking at Dataset 1, our accuracy is quite close to that of the other team.

For future work: We would like to be more nuanced with our feature selection. For simplicity, we mostly just used the top 100 t-test genes. But we could split those up in different ways into different models and combine them into ensembles. This would yield models based on different genes, and hopefully make their predictions more independent of each other. Also, the person who tested models on another dataset, could only get two types of reloaded models to work. So the cross testing of models was more limited than intended.

References

Data:

https://uni-bonn.sciebo.de/s/Uiv84S0XR9XLuch

Background:

https://www.sciencedirect.com/science/article/pii/S2589004219305255?via%3Dihub

Rand Index:

https://www.rdocumentation.org/packages/fossil/versions/0.4.0/topics/rand.index