Phenotypic Prediction from Transcriptomic Features

Vaibhav Mathur, Nittin Aggarwal, Jay Lohokare, Revati Damle

Department of Computer Science

Stony Brook University

Abstract

Motivation: The main aim is to achieve the maximum accuracy in classifying the samples using multi-class classification models.

Results: We could achieve accuracy of 91.31% by using decision tree for predicting population and sequencing center as part of our multi label classification problem. Hence, we were able to conclude multi task learning can reduce the complexity and provide better accuracy for each of the labels.

1 Introduction

In RNA sequencing, the RNA is fragmented, DNA is synthesized complementary to the RNA fragments, which is followed by a complementary strand synthesis. Fragmentation can be done after the cDNA synthesis too. This DNA is then amplified to form a cluster that is sequenced. Most Next-Gen sequencing approaches sequence a short segment of the DNA (it has improved with experimental and technical optimization).

A read refers to the sequence of a cluster that is obtained after the end of the sequencing process which is ultimately the sequence of a section of a unique fragment. If an RNA is expressed in high copies then there will be more reads coming from it; reads can be redundant as well.

Given, output from Salmon, an RNA-seq mapping and quantification tool, on a number of datasets where various samples come from different phenotypes that is cancer populations. The objective is to build a model that takes the Salmon output and predicts the original population class of the unknown genomic data. Features can be selected from the Salmon output, such as TPM, transcript lengths, number of reads and counts. These are provided in the "quant.sf" file. The goal is to achieve the maximum accuracy in classifying the samples using multi-class classification models.

The special properties of this dataset make it difficult to immediately obtain good prediction accuracies by building a well-known classification model like SVM, Random Forest or Naïve-Bayes on it. Each phenotypic category has data points assigned to it (around 20 or fewer samples) while each data point has a huge number of raw features (At least equal to total number of known transcripts for human being in this case which is ~100k, but we have additional features as well). So the main challenge in this problem is selecting and generating few number of descriptive and discriminative features. For that, we have used Decision Trees for dimensionality reduction, after performing statistical analysis on the data.

After feature reduction, we tested various classification models on the reduced dataset. We used models like Random forests, Decision trees, SVM (Linear kernel, RBF Kernel, Poly-kernel). We have verified our model(s) using 5 fold cross-validation, which gave mean accuracy close to 83%.

Furthermore, multi-task learning (predicting more than one label at the same time) reduces the complexity of the model by using more regularization and hence make the model more generalizable which in theory is expected to provide better prediction accuracy for each of the labels. We trained another set of model(s) on a dataset with another label (Sequencing center) in addition to the Population label. We used Decision Tree, Random forests, Binary Relevance based models. The observations of these ex-

periments demonstrated that the accuracy of prediction had increased using the multi-label models compared to the single label models.

2 Methods

2.1 Single Label Classification:

The data-set shared with us consisted of output from Salmon, using which we had to build a model to correctly classify the data. The data set provided was complex and distributed across multiple folders, making it hard to read and aggregate data. Hence, we ran a data-preprocessing step to bring all the data in a single CSV file. We created a data file with every row representing the data corresponding to one sample, having last column of the row as the class of population it belonged to.

By running SVM on the entire raw data-set, we found out that TPM and Effective length were much better parameters to classify the data-set compared to length. We also proved that TPM and Effective length were correlated features, hence we selected only TPM for the further steps in creating a better learning model.

We ran classification algorithms on the dataset to get following results for label prediction experiments –
Decision tree (Average 67%),
SVM (Linear kernel Average 61%),
Random forests (Average 63%).

Though these numbers aren't bad, we realized that accuracy can be increased even further. The data set has multiple (199325) transcripts, all of which may not actually contribute to the classification of population. In-fact, many of these transcripts could potentially be a result of reading errors or noise data that ideally should be filtered out. Hence, there was a need to eliminate the non-contributing transcripts in order to gain a good accuracy of predicting correct label for a given data point. There are multiple approaches for achieving dimensionality reduction of the data set – PCA, Decision trees, High correlation, Low variance and other such statistical tools.

We used decision tree to find out the features that were the most dominant ones in classification of the data set. Surprisingly, out of all the 199325 transcripts, the decision tree returned only 34 features as the ones contributing to population classification. This clearly proves our hypothesis that most of the transcripts in the reads are read errors (noise) and thus completely unrelated to the classification.

After getting these features, we refined our dataset by discarding the other transcripts and their corresponding data points.
We then ran various classification algorithms like SVM, Random Forests, Decision trees on the reduced dataset.

For SVM, we selected gamma value =1/34, and tested using RBF, linear and polynomial kernel(s). To validate our results, we did a 5 fold cross-validation for all models that we used. To further validate the model, we split the data into 2 part (80% + 20%), trained the models using 80% data and then tested the model using the 20% data previously unknown to the model.

Using 5 fold validation on the reduced number of features, we got following results for classification:
SVM (Average 79.9% for linear kernel),
Decision tree (Average 82%),
Random forests (Average 73%).

Apart from the TPM parameters, we also realized that the equivalence classes could contribute to prediction of population. We studied the equivalence classes to find some peculiar properties:

1. Only a few equivalence classes occurred with high frequency. Most others had a very high occurrence rate, and thus could be just some read errors.
2. The number of times a transcript occurs in equivalence classes with higher frequency does bolster its chances of getting selected into the reduced data set.
3. A transcript that occurs more prominently in an equivalence class i.e. it occurs in an equivalence class with lesser number of transcripts also increases the chances of it getting selected to the reduced data set.

We used these observations to find data points to help contribute to the classification problem. We used the sum of weightage of a Transcript in all equivalence classes as a parameter for classification of labels. This 'weightage' of a transcript in an equivalence class can be found by dividing the number of reads of the equivalence class by the equivalence class size.

Sum of all such weightages gives the overall importance of a transcript based on the equivalence class data. However, we can make this more efficient by pruning the equivalence classes – select the weightage only from those equivalence classes that have read count greater than a threshold.

2.2 Multi Label Classification:

Multi-label classification on the dataset is a promising way to improve the overall accuracy of prediction of a label. One approach to Multi Label classification is to ensemble two models which classify the labels individually. Another approach was to use Classification models that give predictions for multi-dimensional input (multiple labels). We used Decision Tree and Random forest classifiers to classify the 2-label data. We also built a model using the Binary relevance method.

Calculation of accuracy of predictions in Multi-Label classification can be done in multiple ways. One way is to consider only the outputs with both labels correctly predicted as a correct prediction. However, this is highly stringent criteria as it doesn't give positive weightage to outputs with one label correctly predicted. The approach we used involves giving 0.5 weightage to outputs with single label correctly predicted and 1 weightage to outputs with both labels correctly predicted.

3 Results

Decision Tree used to reduce dimensions, then using SVM, RF, DT as classifiers. (Using TPM)
Initial shape of dataset: (369, 199325)
Size of data after feature reduction using decision tree: (369, 34)

Decision Tree 5 Fold Results:
Scores: [0.79220779 0.85526316 0.86111111 0.77777778 0.83333333]
Mean: 0.823938634465
F1 Scores: [0.81956486 0.83837535 0.83079273 0.76363937 0.84381808]
F1 Mean: 0.819238078988

SVM 5 Fold Results for different kernels :
Linear:
Scores:[0.83116883 0.75 0.84722222 0.77777778 0.79166667]
Mean: 0.799567099567
F1 Scores:[0.83083862 0.75444085 0.84771659 0.774364 0.79009025]
F1 Mean:0.799490063085
Poly:
Scores: [0.7012987, 0.63157895, 0.73611111, 0.73611111, 0.75]
Mean: 0.711019974178
F1 Scores: [0.70061622, 0.6389158, 0.74710089, 0.732282, 0.74301994]
F1 Mean: 0.712386970787

RBF:
Scores: [0.41558442 0.48684211 0.40277778 0.375 0.375]
Mean: 0.411040859725
F1 Scores: [0.40318841 0.46642951 0.42463092 0.38231183 0.36566169]
F1 Mean: 0.408444472578

A. Random Forest 5 Fold Results:
Scores:[0.81818182 0.64473684 0.73611111 0.73611111 0.75]
Mean:0.737028176502
F1 Scores:[0.74594982 0.77619359 0.79129191 0.73495214 0.80150312]
F1 Mean:0.769978117359

Results without reduction, using SVM, RF, DT classifiers:
Initial shape of dataset: (369, 199325)

Decision Tree 5 Fold Results:
Scores:[0.64935065 0.65789474 0.66666667 0.68055556 0.72222222]
Mean:0.675337966127
F1 Scores:[0.68293423 0.67138004 0.70656977 0.73974451 0.70547788]
F1 Mean:0.701221287246

SVM 5 Fold Results for kernel :
linear
Scores:[0.61038961 0.57894737 0.625 0.61111111 0.65277778]
Mean:0.61564517354
F1 Scores:[0.62019704 0.5780541 0.63516484 0.61221376 0.6576092]
F1 Mean:0.620647785786
poly
Scores:[0.61038961 0.55263158 0.66666667 0.61111111 0.65277778]
Mean:0.618715348979
F1 Scores:[0.60658785 0.54944241 0.6700062 0.62020942 0.65634281]
F1 Mean:0.620517737211
rbf

Scores:[0.20779221 0.22368421 0.23611111 0.20833333 0.25]
Mean:0.225184172553
F1 Scores:[0.1017316 0.11805627 0.12179487 0.11293341 0.15277931]
F1 Mean:0.121459090555

Random Forest 5 Fold Results:
Scores:[0.62337662 0.59210526 0.72222222 0.625 0.61111111]
Mean:0.634763043974
F1 Scores:[0.68338429 0.55754579 0.68796512 0.6925062 0.55393413]
F1 Mean:0.635067106769

Using 80% of data for training and 20% as unknown for testing:
Number of rows: 295
Size of data after feature reduction using decision tree: (295, 27)

Decision Tree 5 Fold Results:
Scores:[0.86666667 0.79661017 0.83050847 0.84745763 0.82758621]
Mean:0.83376582895

SVM 5 Fold Results: (linear kernel)
Scores:[0.7 0.71186441 0.76271186 0.76271186 0.77586207]
Mean:0.742630040912

Random Forest 5 Fold Results:
Scores:[0.81666667 0.72881356 0.76271186 0.81355932 0.84482759]
Mean:0.793315799727

Predicting over the remaining 20% data
Accuracy using Decision Tree on the Test Data:
0.648648648649

Confusion Matrix for Decision Tree on the Test Data:
[[11 3 0 2 0]
[1 8 1 1 1]
[1 4 11 2 0]
[0 3 3 4 2]
[1 1 0 0 14]]

Accuracy using SVM on the Test Data:
0.77027027027
Confusion Matrix for SVM on the Test Data:
[[15 1 0 0 0]
[0 11 0 1 0]
[0 1 13 4 0]
[0 2 2 8 0]
[2 2 1 1 10]]

Accuracy using Random Forest on the Test Data:
0.783783783784

Confusion Matrix:
[[15 1 0 0 0]
[2 10 0 0]
[0 2 14 2 0]
[1 1 3 6 1]
[0 2 1 0 13]]

We found that running the decision tree classifier multiple times in order to select the features, we get different set of features (But the count is same). The initial features (root and its immediate children) are same in all the results, but the nodes in the lower part of the tree keep changing. We think that this occurs due to multiple correlated features existing in the data set - The decision tree selects one of these features randomly.

Results from the equivalence class feature:

Initial shape of dataset:
(369, 199325)
Size of data after feature reduction using decision tree : (369, 29)

Decision Tree 5 Fold Results:
Scores:[0.8961039 0.89473684 0.70833333 0.90277778 0.80555556]
Mean Accuracy:0.841501480975
F1 Scores:[0.89586022 0.85817597 0.73045334 0.94471058 0.79218179]
F1 Mean:0.844276380522

Random Forest 5 Fold Results:
Scores:[0.79220779 0.73684211 0.81944444 0.84722222 0.84722222]
Mean Accuracy:0.808587757272
F1 Scores:[0.78150538 0.78263597 0.79452808 0.75805596 0.81359513]
F1 Mean:0.786064103686

Results from the Multi Label Classification model for predicting both Population and Sequencing center:
Using 70% of data to train and 30% to test:
Initial Size of data:
(369, 199326)
Size of data after feature reduction using decision tree : (369, 59)
Accuracy using Decision Tree on the reduced feature set: 0.910472972972973
5 Fold Mean Accuracy: 0.878063164
Accuracy using RF on the reduced feature set: 0.9341216216216216

5 Fold Mean Accuracy: 0.84410924

Accuracy using Binary Relevance on the reduced feature set:
0.8614864864864865

Results for predicting only Sequencing center using Random Forest and Decision Tree as classifiers:

Initial shape of dataset:
(369, 199326)
Size of data after feature reduction using decision tree : (369, 16)

Decision Tree 5 Fold Results:
Scores:[0.92105263 0.97333333 0.93150685 0.87671233 0.95833333]
Mean:0.932187695266
F1 Scores:[0.91920123 0.96146677 0.90738019 0.88822824 0.94505675]
F1 Mean:0.924266635904

Random Forest 5 Fold Results:
Scores:[0.96052632 0.96 0.93150685 0.91780822 0.98611111]
Mean:0.951190499079
F1 Scores:[0.91559055 0.97948718 0.90457092 0.92314941 0.96336225]
F1 Mean:0.937232061705

Acknowledgements

This work would not have been possible without the consistent and kind support from Ms Fatemah Almodaresi. We are indebted towards her valuable insights and prompt explanations. We would also be not able to proceed in the project without foundational knowledge imparted with careful brevity from our Professor Rob Patro.

References:

1. http://salmon.readthedocs.io/en/latest/index.html

2. http://scikit.ml/api/classify.html#

3. http://scikit-learn.org/stable/modules/tree.html

4. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html