Phenotypic Prediction from Transcriptomic Features

Arjun Mathew Dan, Ajin Oommen Kuriakose, Manu Mathew, Shilpa Mary George

***Abstract*—A central problem in machine learning is identifying a representative set of features from which to construct a classification model for a particular task. We are provided with output from Salmon, an RNA-seq mapping and quantification tool. The various samples come from different phenotypes; types of population in this case. Our goal is to perform feature selection/reduction and to come up with a model that takes the Salmon output and predicts the original label and to achieve the maximum accuracy in classifying the samples using multi-class classification models.**

***Index Terms*—Transcript Quantification, TPM, Prediction, Salmon**

I. Data Format

The train data was taken from the provided "quant.sf" file which is the output from Salmon. There are 369 samples each having around 200k features. This data was formatted by performing data flattening such that each row corresponds to a sample having the entire feature list. The final data format on which we have build our model is depicted in the Figure below.

A. Ambig Info

Apart from "quant.sf", we have considered the auxiliary file "ambig_info.tsv" as well. This file contains information about the number of uniquely-mapping reads and the total number of ambiguously-mapping reads for each transcript. Both "AmbigCount" and "UniqueCount" from this file were added as additional features.

B. Equivalence Class

The top 100 Equivalence classes were selected from "eq_classes.txt" on the basis of the count of fragments.
Transcripts belonging to a particular equivalence class were assigned with the fragments count value corresponding to that equivalence class.
If a higher fragments count is observed in any other equivalence class where the feature is a member, its value is updated with this higher value.

After processing the selected equivalence classes, Transcripts would get the best possible fragment value associated with it. Transcripts which were not there in the equivalence class would get a value of 0. This value would be appended with "quant.sf" data along with "Ambig_info" data.
The above features were combined to generate a new file "quant2.sf" which is then used to train the model (using run.sh file).

Name	ENST00000456328.2					
	EffectiveLength	TPM	NumReads	UniqueCount	AmbigCount	Eq_Class
0	1322.77	0.000000	0.00000	0.0	1.0	0.0
0	182.00	0.000000	0.00000	0.0	0.0	0.0
0	1396.78	0.000000	0.00000	0.0	2.0	0.0
0	175.00	0.000000	0.00000	0.0	0.0	0.0
0	1423.57	0.148344	5.35495	2.0	10.0	0.0

Sample data format after including ambig_info and eq_classes

C. Labels

The input data provided had two labels, namely, **population** and **sequencing center**. To facilitate multi-task prediction, we created a new label **combined**, by merging these two labels.

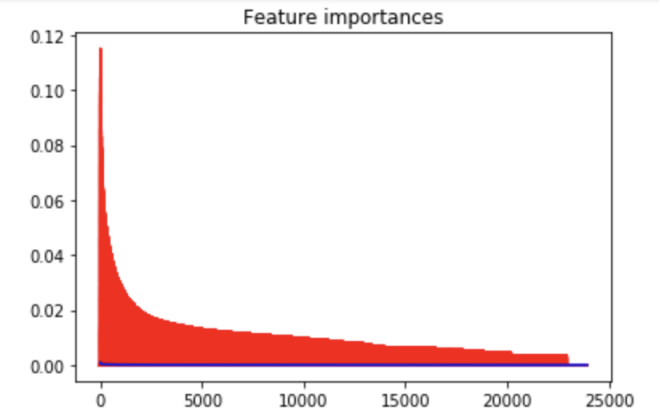
The data and labels are randomly shuffled (while maintaining their relative order).

II. Feature Selection

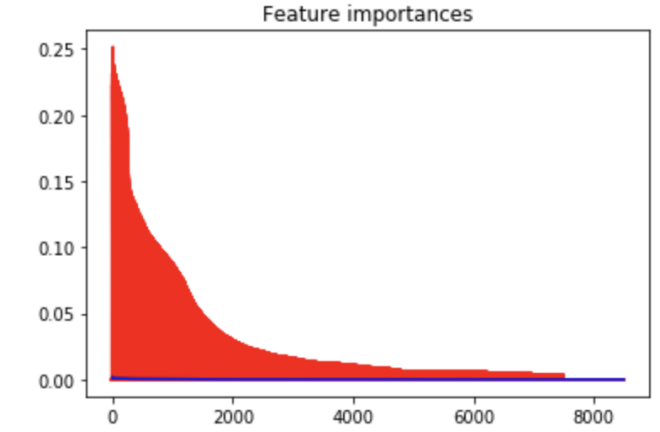
Initially upon analyzing the feature set, we found that there is a co-relation among the features provided i.e, between Length and Effective length; therefore we omitted the Length.

A. ExtraTreesClassifier

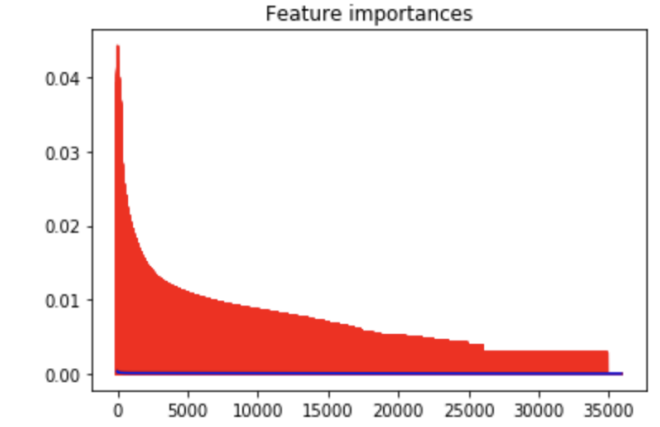
We used ExtraTreesClassifier for performing feature selection. Feature selection was performed independently for each of the three classifiers that we modelled : i.e for "population" label, "sequencing center" label and "combined" label. Using the ExtraTreesClassifier we were able to figure out the top relevant features for each chosen model. The data set was then processed to remove the features that were not important . This process significantly reduced the feature dimension for the model. The feature importance for each of the classifiers is substantiated by plotting the feature importance against the count of the features for each classifier as shown below(considering the entire train data).



The feature importance for ”population” label



The feature importance for ”sequencing center” label



The feature importance for ”combined” label

The best features were selected on the basis of the above observations.

B. Other Approaches Tried - Dimensionality Reduction

We used Principal Component Analysis(PCA) to change the feature domain and map the input feature list to an enhanced set of features. The data was transformed using the enhanced feature set from PCA.The model was trained using this transformed data set as training input. However, we got better results with Feature Selection and hence this approach was discarded.

III. Classifier Models

Different Classifier models such as Random Forest and Logistic Regression and Neural Nets were tried. Among these, we observed better results with Random Forest.

A. Cross Validation

After selecting the classifier model, we performed three-fold cross validation (3-fold due to a small data size). During cross-validation, feature selection was done after creating the fold on the training data and then this data was used to train the model. Features selected were then used to select the best features in the test data. The above cross validation run was independently performed for each of the three classifiers that we modelled : i.e for ”population”, ”sequencing center” and ”combined” labels .

The 3 fold Cross Validation results for ”population” label is depicted below. The result shows the precision recall, f1-score and support for each of the classes and for average total for the model:

```
Performing cross validation for label :  population
Train data shape (246, 15990) label shape (246,) test shape (123, 15990)
             precision    recall  f1-score   support

        CEU       0.80      1.00      0.89        16
        FIN       0.80      0.86      0.83        28
        GBR       0.81      0.81      0.81        31
        TSI       0.77      0.59      0.67        29
        YRI       0.95      1.00      0.97        19

avg / total       0.82      0.82      0.81       123

Train data shape (246, 15841) label shape (246,) test shape (123, 15841)
             precision    recall  f1-score   support

        CEU       0.83      1.00      0.91        24
        FIN       0.86      0.83      0.85        30
        GBR       0.94      0.81      0.87        21
        TSI       0.70      0.67      0.68        21
        YRI       0.96      0.96      0.96        27

avg / total       0.86      0.86      0.86       123

Train data shape (246, 16079) label shape (246,) test shape (123, 16079)
             precision    recall  f1-score   support

        CEU       0.97      0.91      0.94        32
        FIN       0.81      0.89      0.85        19
        GBR       0.85      0.88      0.86        25
        TSI       0.77      0.77      0.77        22
        YRI       1.00      0.96      0.98        25

avg / total       0.89      0.89      0.89       123
```

Similarly, the best Cross Validation result for ”sequencing center” label is depicted in the below figure:

```
Performing cross validation for label :  sequencing_center
Train data shape (246, 5949) label shape (246,) test shape (123, 5949)
             precision    recall  f1-score   support

          1       0.97      1.00      0.98        32
          2       1.00      1.00      1.00        30
          3       1.00      1.00      1.00        12
          4       1.00      0.95      0.97        20
          5       1.00      1.00      1.00        11
          6       1.00      1.00      1.00         9
          7       1.00      1.00      1.00         9

avg / total       0.99      0.99      0.99       123
```

And the best Cross Validation result for "combined" label is depicted in the figure below:

```
Performing cross validation for label :  combined
Train data shape (246, 24387) label shape (246,) test shape (123, 24387)

             precision    recall  f1-score   support

       CEU1       0.67      1.00      0.80         4
       CEU2       1.00      0.75      0.86         4
       CEU3       1.00      0.60      0.75         5
       CEU4       1.00      0.14      0.25         7
       CEU6       0.00      0.00      0.00         1
       CEU7       0.00      0.00      0.00         5
       FIN1       0.43      0.60      0.50         5
       FIN2       0.75      0.60      0.67         5
       FIN3       1.00      0.33      0.50         6
       FIN4       0.44      1.00      0.62         4
       FIN5       0.00      0.00      0.00         1
       FIN6       0.00      0.00      0.00         1
       FIN7       0.00      0.00      0.00         1
       GBR1       0.67      0.29      0.40         7
       GBR2       0.40      0.50      0.44         4
       GBR3       0.50      0.40      0.44         5
       GBR4       0.20      0.50      0.29         2
       GBR5       0.20      0.50      0.29         2
       GBR6       0.60      1.00      0.75         3
       GBR7       0.50      1.00      0.67         2
       TSI1       0.50      0.75      0.60         8
       TSI2       0.67      0.40      0.50         5
       TSI3       0.00      0.00      0.00         2
       TSI4       0.50      0.20      0.29         5
       TSI5       0.50      0.50      0.50         2
       TSI6       0.00      0.00      0.00         2
       TSI7       0.00      0.00      0.00         0
       YRI1       0.50      0.43      0.46         7
       YRI2       1.00      1.00      1.00         5
       YRI3       0.29      0.50      0.36         4
       YRI4       0.00      0.00      0.00         1
       YRI5       1.00      0.25      0.40         4
       YRI6       0.33      1.00      0.50         1
       YRI7       1.00      0.33      0.50         3

avg / total       0.59      0.48      0.47       123
```

B. Final Model

After selecting the best model to use after the cross validation process and fixing the hyper-parameters , we trained the model on the entire training data set (369 samples) . Feature selection was done considering the entire samples and the model is build on top of the dimensionality reduced data. The trained model along with the importance feature indices are then together picked to a single file for each of the classifier. Each of this model is stored in a file with the format - model_classifier_name.pkl , which is used during the testing process.

IV. Results and Findings

- We observed that removing the length attribute from each of the feature didn't affect the overall accuracy.
- Adding 2 additional attributes from Ambiginfo file resulted in an improved accuracy. For instance training the model using 300 samples without including attributes from "Ambiginfo" gave an accuracy of 0.78 in comparison to 250 samples with "Ambiginfo" attributes included giving an accuracy of 0.81. This was our **Intermediate result**
- Performing **feature selection** using ExtraTreeClassifier gave an improvement on the accuracy by another 5 percent. Here, only the best features for each model was chosen.
- Adding the data from the **Equivalence Classes** further improved the accuracy by around 3 percent. Thus the overall best accuracy of our Model for the cross validation runs is **0.89** for the 'Population Classifier', **0.99** for the 'Sequencing Centre classifier' and **0.59** for the combined multi-task classifier.
- In our experiments **Random Forest** gave a comparatively better accuracy when compared to other classifiers that we have tried.
- Shuffling the data initially and pre-processing it by normalization and standardization also slightly improved the accuracy of our model

V. Steps to run

We are performing some pre-processing on the input data to add additional features from the ambig-info and equivalence class files. As these information is contained in separate files for each sample, we are using a bash script to pre-process the data.Inside the script, a new file "quant2.sf" file is generated after processing "ambig_info.tsv" and "eq_classes.txt" files from aux_info directory for every sample. More details about how we use equivalence class is mentioned in the corresponding section.
These parameters are then passed to the python executable file for the prediction.
Note: This has been executed and verified on Mac Linux OS only.
In order to test the model, *"run.sh"* bash script has to be invoked with the required arguments. Usage is given below.

./run.sh -m "location of the model dump" -t "location of the test samples directory" -l "location of the test labels file"
Wrapper shell script, "run.sh" takes the following arguments.

- Model dump directory path
  This directory contains the pickled model dump.
- Test sample directory path
  This is the directory where all the test samples are present.
- Path to test label file

E.g. *./run.sh -m /Desktop/model/ -t /Desktop/train/ -l /Desktop/p1_train_pop_lab_test_label.csv*

```
Please wait...
....

F1 Score and Accuracy for Classifier label - Population

             precision    recall  f1-score   support

        CEU       1.00      1.00      1.00         2
        FIN       1.00      1.00      1.00         1
        GBR       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00         4

F1 Score and Accuracy for Classifier label - Sequencing Centre

             precision    recall  f1-score   support

          1       1.00      1.00      1.00         1
          2       1.00      1.00      1.00         1
          7       1.00      1.00      1.00         2

avg / total       1.00      1.00      1.00         4

F1 Score and Accuracy for Classifier label - Joint

             precision    recall  f1-score   support

       CEU2       1.00      1.00      1.00         1
       CEU7       1.00      1.00      1.00         1
       FIN7       1.00      1.00      1.00         1
       GBR1       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00         4
```

Sample output of running the script on the same data set as used for *training*

VI. References

- http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html
- http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- http://salmon.readthedocs.io/en/latest/file_formats.html