Uncertainty Quantification

Iti Inani – 111323922, Venkata Sainath Chaluvadi – 111498013,
Vishwatej Reddy – 111446995, Akshat Singhal – 111496103

Department of Computer Science, Stony Brook University, Stony Brook, New York-11790

Abstract:
Motivation:Salmon is a tool that determines expression levels by solving a maximum likelihood problem. A result of this formulation is that one often gets accurate estimates of the transcript abundances, but has no notion of confidence in these predictions. That is, predictions can be highly specific and highly accurate, or highly specific and highly inaccurate --- this depends on the "shape" of the likelihood function, and how optimization proceeds. Thus, it is very valuable to provide some measure of confidence in the estimates that are inferred.

1 Introduction

The goal of this project is to provide some measure of confidence in the estimates that are inferred by this formulation. To achieve this, we are analyzing Salmon's bootstrapping output and filtering the failing transcripts and then finding some common properties between them, using transcript quantification.
We identified faulty transcripts using bootstrap outputs and poly_truth output file. We analyzed the list of faulty transcripts and tried to provide some statistics regarding their common properties. We are using SVM classification model[2] to train the data and predict the faulty transcripts based on the common properties.
After classification we are using regression model to predict the error in faulty transcripts.The predicted error indicates the deviation of numReads value of faulty transcripts from the truth value.Our goal is to reduce the number of faulty transcripts in the dataset. Predicted error is used to change the values in quant_bootstrap.tsv in such a way that reduces the count of faulty transcripts.

2 DataSets

We are provided following datasets as the input-
a) Quant.sf - Salmon quantification file that contains following information about each transcript: Name, Length, Effective length, TPM, NumReads[1].
b) eq_classes.txt - Set of equivalence classes + their counts and transcript weight distributions.
c) poly_truth.tsv – true transcripts NumReads.
d) quant_bootstraps.tsv – bootstrapping output of Salmon.

3 Solution Methodology

3.1 Identifying faulty transcripts

A transcript is considered faulty if its truth value is not within the 95% confidence interval.95% confidence interval is the range from [mean - 2 * standard deviation] to [mean + 2 * standard deviation], where mean and standard deviation are calculated over the bootstrap samples.
For the given datasets of 93109 transcripts, we found the count of faulty transcripts as:
poly_mo – 13766
poly_ro - 22710

We did some extra analysis to identify pattern in deviation of mean reads value. We analyzed our training dataset's truth value and error value and calculated the error fraction with the formula-

$$err(T) = mean(T) - trucount(T)/\ mean(T)$$

We didn't use this error fraction as a feature or parameter in any of our training model since truth values cannot be used.But for analysis and finding the pattern we used it to create following graph.
It can be observed that maximum transcripts lie in the range from 0 to 0.5 that means mean numReads is greater than truth value most of the times.

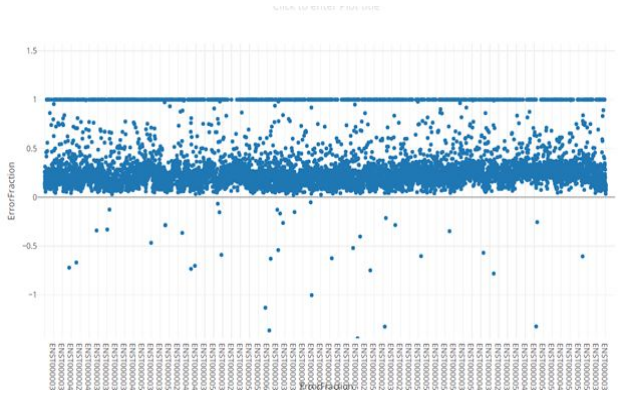


Fig 2. y-axis:-Error fraction of each transcript, x-axis: transcript ID.

3.2 Establishing common properties

Since, the input data contains only quantification estimates, we analyzed:
1) equivalence classes
2) estimated TPM values
3) count of ambiguous vs uniquely mapped reads
4) true read count of each transcript

Based on these parameters like equivalence classes , we came up with following 3 properties which were able to predict faulty transcripts with about 90% accuracy:
a) **Weight:** Based on the occurrence of the transcripts in the equivalence classes, we assigned weights to the transcripts. 1 for the transcripts that occur only in one of the equivalence class and 0 for the transcripts that are occurring in multiple equivalence classes.
b) **Uniquely mapped:** Based on the above weights, we introduced a new property that will be able to differentiate the uniquely mapped transcripts from the ambiguously mapped transcripts.
c) **Estimated TPM value** - Additionally, since the TPM value for each transcript is directly proportional to read count and inversely proportional to its length, TPM becomes a very intuitive choice.

3.3 Feature Selection

Property	Usage	Description
Length	-	As there was not much difference between the values of faulty and non-faulty transcripts with respect to this property, this is ignored during the classification of faulty transcripts.
Effective Length	-	As there was not much difference between the values of faulty and non-faulty transcripts with respect to this property, this is ignored during the classification of faulty transcripts
true read count		
TPM	Classification and Regression	Since the TPM value for each transcript is directly proportional to read count and inversely proportional to its length and also because it contains information about length and count together, TPM becomes a very intuitive choice.
NumReads	Classification and Regression	total count of reads mapped to a transcript (either uniquely mapped to that transcript, or mapped to that transcript and other transcripts as well which makes it ambiguous mapping as 100% of this read cannot be counted for a specific transcript.) And also, as these values close to the actual truth values, this was introduced as one of the properties that can be used to identify the faulty transcripts.
Weight	Regression	Based on the occurrence of the transcripts in the equivalence classes, we assigned weights to the transcripts
Uniquely mappable	Classification and Regression	If all the reads assigned to a transcript are uniquely mapped, we call the transcript uniquely mappable.
Error Fraction	-	We calculate error fraction as the difference of true read count and estimated read count divided by true read count i.e., $\frac{(trueNumReads - estimatedNumReads)}{trueNumReads}$ This would have been an ideal parameter to classify, but as there would not be any truth values available for the new dataset, this was ignored.
Mean	Regression	In order to decrease the number of faulty transcripts, we calculated a new mean such that the values after the change lies within the 95% confidence interval (i.e. greater than at least 2.5% of the bootstrap samples and less than the upper 97.5 percentile of the bootstrap samples)
Variance	Regression	As standard deviation values are very small, we used variance to make it more useful in identifying the

3.4 Creating the Classification model and predicting faulty transcripts.

a) **Dataset and features for classification** – We used complete dataset provided to us(poly_mo and poly_ro) as training dataset.
b) **Classification Model** – We are using Support vector machine(SVM) to classify the dataset using python's sklearn library[2].
c) **Prediction** -Testing dataset was passed to the classification model to predict the faulty transcripts.The prediction function returns the list with values 0 or 1. Size of the list is the total number of transcripts.
Each value of the list indicates if the transcripts is faulty(1) or non-faulty(1).

3.5 Creating the Regression model to predict error value.

a) **Dataset and features for regression -** Input dataset for training the regression model is the faulty transcripts list and its properties.
We are filtering the faulty transcripts and using this as the input because this model will be used to predict the error value. (deviation of mean num reads from true read count).
b) **Regression Model -** We are using a linear regression model to predict error values.
c) **Predicting error values -** Testing dataset which contain filtered faulty transcripts is passed to the regression model which then predicts the **Error value.** This error value is the difference between numReads (mean of bootstrap result) and true count(in poly_truth.tsv).

3.6 Changing the quant_bootstrap.tsv

To reduce the number of faulty transcripts we are using the predicted error values to change the mean of numReads of faulty transcripts to reduce its deviation from true count.
We had two options to change the mean value -

1. Reduce the mean value - when error value is negative, this indicates truth value would have been lesser than mean and beyond the lower bound(-2σ) of our confidence interval.
2. Increase the mean value - when error value is positive, this indicates truth value would have been greater than mean and beyond the upper bound(2σ) of our confidence interval.

We followed following algorithm to change the values in quant_bootstrap.tsv-

```
For each column in quant_bootstrap.tsv
    If transcript ID  is in list of faulty transcripts
        For each row:
          If predicted_error < 0:
              new_value = old_value -  predicted_error
          else:
              new_value = old_value  + predicted_error
    End If.
End For
```

Using above algorithm, each cell value for faulty transcripts in bootstrap file will be changes. This affects the mean reads for the transcripts. As the mean value increases , this in turn increases the chances of truth value to lie in the confidence interval. Hence reducing the difference between mean and true count.

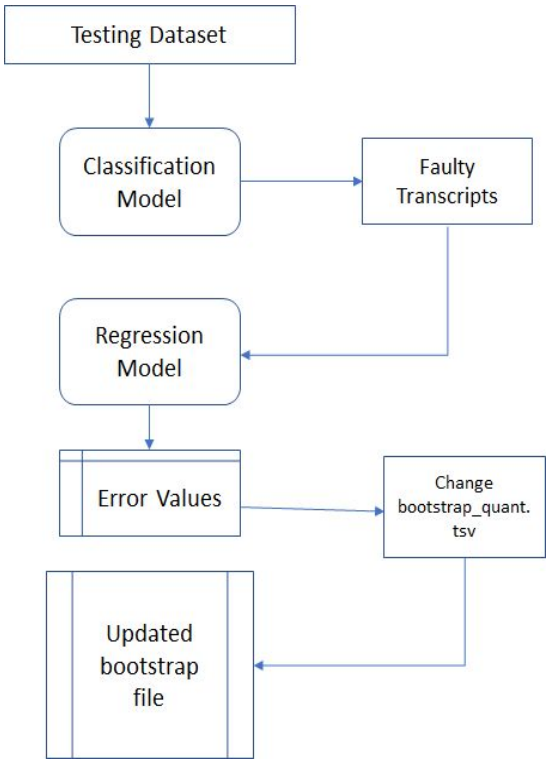


Fig 1 - Diagram for complete Procedure for testing dataset and reducing faulty transcript counts from bootstrap file.

4 Results-

4.1 Faulty transcripts prediction using single dataset

Initially we tested each dataset individually. Used the same dataset for training and testing by splitting the dataset and using cross validation to analyze the accuracy.
Both poly_mo and poly_ro were tested individually and we got following result-

We analyzed the accuracy of our prediction output with the actual output using the confusion matrix.
There are 4 values in confusion matrix, each of which are described below:
1) True Negative – Actual non-faulty transcripts predicted as non-faulty (top-left) (6879)
2) False Positive – Actual non-faulty transcripts predicted as faulty. (top-right) (159)
3) False Negative – Actual faulty transcripts predicted as non-faulty. (bottom-left) (520)
4) True Positive – Actual faulty transcripts predicted as faulty. (bottom-right) (1753)
Below is the output of 1 execution for dataset 'poly_ro'.

```
[[6879  159]
 [ 520 1753]]
             precision    recall  f1-score   support

      False       0.93      0.98      0.95      7038
       True       0.92      0.77      0.84      2273

avg / total       0.93      0.93      0.92      9311
```

Here, we have achieved 93% accuracy in identifying non-faulty transcripts (row 1, column 1) and 92% accuracy in identifying faulty transcripts (row 2, column 1).
Currently, our model can predict faulty transcripts with a precision in the range of 89% - 93% for dataset 'poly_ro' and 87% - 91% for dataset 'poly_mo' and non-faulty transcripts with an accuracy range of 93% - 96%.

4.2 Faulty Transcript Prediction using poly_mo dataset-

For increase the size of our dataset we decided to use complete poly_mo and poly_ro dataset for training the model for classification and also for training the model for regression.The advantages of this approach are -

1. Large amount of data leads to better training of the model. Hence will lead to better prediction.
2. Testing would also be performed on complete dataset. External dataset that would be input to this application might be similar to the poly_mo dataset and size will also be similar,hence will share some common characteristics. This increases our chances of accurate predictions.

We used complete poly_mo dataset and poly_ro dataset for training the model and then passed poly_mo as testing dataset to this application-
We received following results-

```
[[77349  1994]
 [ 4334  9432]]
             precision    recall  f1-score   support

      False       0.95      0.97      0.96     79343
       True       0.83      0.69      0.75     13766

avg / total       0.93      0.93      0.93     93109
```

Since we increased the size of out dataset and instead of testing just on a part of dataset, we are testing it on a complete data, precision dropped for identifying faulty transcripts.9432 faulty transcripts were predicted correctly and we further processed these transcripts in the regression model

4.3 Error value prediction using regression model(poly_mo dataset)-

Faulty transcripts predicted in classification model were then passed to regression model to predict the error values.
R2 score for poly_mo dataset - 0.84
We further used these error values to update the input quant_bootstrap.tsv.
To verify if the faulty transcripts are reduced,we again followed the procedure of filtering faulty transcripts using mean and standard deviation for confidence interval of 95%(Step 1).

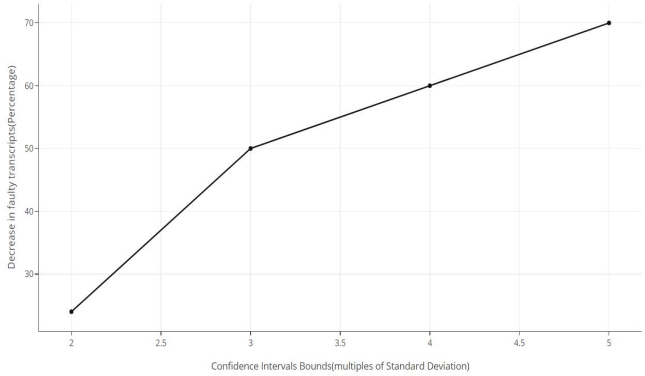
Initial faulty transcripts count - 13766
Reduced faulty transcripts count(using this application) - 10614

After performing this experiment multiple times we got an average **24%** reduction in count of the faulty transcripts.

4.4 Further Analysis-

We further analyzed on how can the count of faulty transcripts can be reduced.We observed that if we change confidence interval to
new_mean = $\pm 4\sigma$ percentage reduction = 60%
new_mean = $\pm 5\sigma$ percentage reduction = 70%

Since the value of Standard Deviation(σ) is considerably smaller than the mean. So increasing the mean in steps of standard deviation was not shifting the graph significantly.
So if we want to reduce the count further,it would be better instead of just shifting the mean if we could spread the graph and increase standard deviation value.
Below is the graph that shows the decrease in faulty transcripts count with increase in standard deviation value.



5 Challenges Faced/Limitations-

Our application uses regression model that has been trained on poly_mo dataset.
When a dataset similar to poly_mo is tested against it ,the predicted error values have r2 score 0.84 which is helping in reducing the faulty transcripts count.
But if we test dataset poly_ro with the same training model,the r2 score is negative.
We looked at the bootstrap file,mean and standard deviation of both the dataset

6 References

[1] Salmon Output Files-
http://salmon.readthedocs.io/en/latest/file_formats.html#
[2] SKLearn Library SVM classification model -
http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[3] Course Page Projectlist:
https://docs.google.com/document/d/1QiIOegxuSYQxfEB7W6qb8KWvzqFHOTdG6TUVhPveEnw/edit