RNA-Seq Analysis of Latent Tuberculosis CD4 T-Cells and Comparison of Predictive Classifier Models

Walker Azam

Introduction

Latent Tuberculosis Infection (LTBI) is an inactive form of Tuberculosis (TB), wherein the TB bacteria resides within a body in an inactive state, and the patient shows no symptoms. However, this can progress to active TB, specially within people with weakened immune systems. It remains an important challenge to find effective ways to detect and screen for LTBI, as well as develop treatments[2].

Within the scope of this project, a dataset of Latent TB within Human CD4 T-Cell samples will be used to identify a set of Differentially Expressed Genes (DEG) that can be used in further downstream analysis as genes of interest in treating LTBI. This set of DEGs could be used to inform future research, and find enriched GO pathways.

Moreover, common machine learning models will be compared to identify effective classifiers in predicting Latent TB. Although there are existing options to test patients for latent TB[2], an accurate predictive model would be beneficial in drug development and testing, potentially lowering costs associated with checking treatment efficacy.

Dataset

The dataset is from the GEO database, hosted by the NCBI, from a study by Burel et al., 2018 (GSE99373)[1]. The study design performed RNA-sequencing via Illumina HiSeq 2500 on 39 Human subject's memory CD4 T-Cells. 20 of these subjects had latent TB, whereas 19 were healthy controls. The raw counts included 21,920 genes for these 39 samples, and run metadata was available on the GEO Accession Display.

Methods

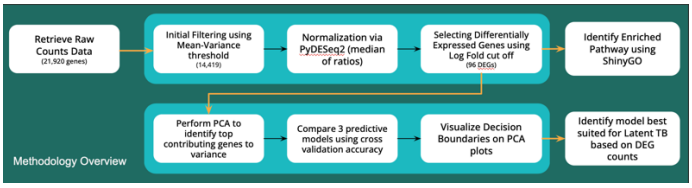


Figure 1: Methodology Workflow

Fig. 1 provides an overview of the methodology workflow used in this project. All data processing, modeling, and visualization was done using Python 3.10. Raw data was first filtered using a mean-variance threshold to reduce the initial number of genes based on information loss. A median-of-ratios method was used to normalize the remaining gene counts, using PyDESeq2[3]. DEGs were then identified using an absolute Log2 Fold Change of at least 1 among gene expression, and a Benjamin-Hochberg adjusted p-value < 0.05. ShinyGO[4] was used to identify any enriched KEGG pathways using this set of DEGs (**Table 1**).

Enrichment FDR	nGenes	Pathway Genes	Fold Enrichment	Pathways (click for details)
3.2E-02	3	67	19.3	Acute myeloid leukemia

Table 1: ShinyGO[4] Output for Enriched KEGG Pathways

PCA was performed on standardized Log2-transformed counts for the DEGs, where the top 5 genes associated with the first two Principal Components were identified (42.4% of total variance [**S.Fig 2A**]). 3 common predictive models – Naive-Bayes (NB), Support Vector Machine (SVM), and Nearest Neighbor (KNN) – accuracies were compared using a 5-fold Cross Validation. Decision Boundaries were visualized upon the PCA plots to better understand and compare model performances. Most default priors were kept, and SVM used a Radial kernel, and KNN used a value of *K=2*. Models were all from the Scikit-Learn Python library.

Results

Differentially Expressed Genes (DEGs)

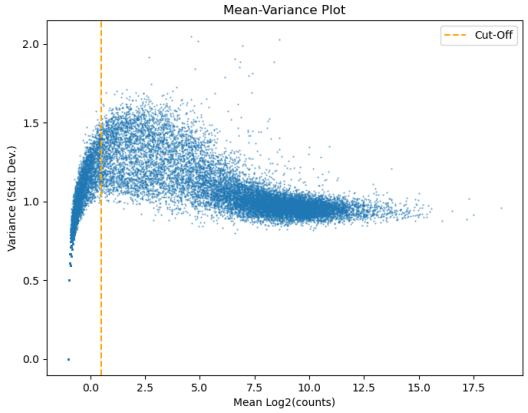


Figure 2: Mean-Variance Plot

Selecting for the DEGs first involved filtering. **Fig 2** shows a Mean-Variance plot, where genes' mean Log2 expression value is plotted against its Variance. There is a sharp 'hook' at the mean Log2 value of 0.5. This indicates information loss in genes past this point - where there is both low mean and low variance. These genes are mostly undetected, or would confer little meaning in downstream analysis, and could then be filtered out. This results in 14,419 remaining genes, which were normalized to find the list of 96 DEGs used for enrichment analysis. **Table 1** shows that this list only found 1 significantly enriched pathway (Acute Myeloid Leukemia) indicating the selection process may have been too stringent for enrichment analysis. However, it could be useful for modeling, which is what was considered next.

Model Performances

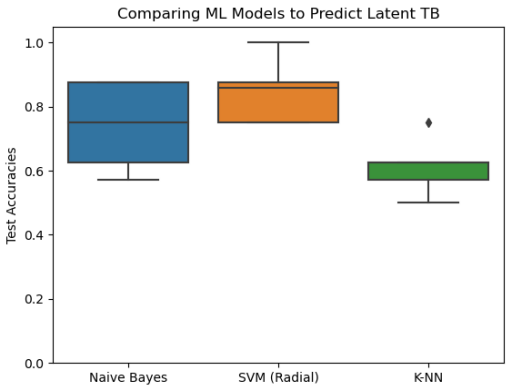


Figure 3: Predictive Model Cross Validation Results

Using the Log2-transformed counts for the 96 DEGs, three different predictive classifiers were compared with a 5-fold cross-validation across all 39 samples. **Fig 3** shows that the SVM model fit with a radial kernel had the highest average test accuracy (84.6%), whereas Naive-Bayes had 73.9% accuracy and KNN had 62.4% mean accuracy. Based on the small number of samples, a simpler heuristic model such KNN or Naive-Bayes may have been assumed to be more accurate, however **Fig 3** shows that SVMs perform best. This also demonstrates that the selected DEGs serve as a good feature set to train models for LTBI prediction.

Model Decision Boundaries

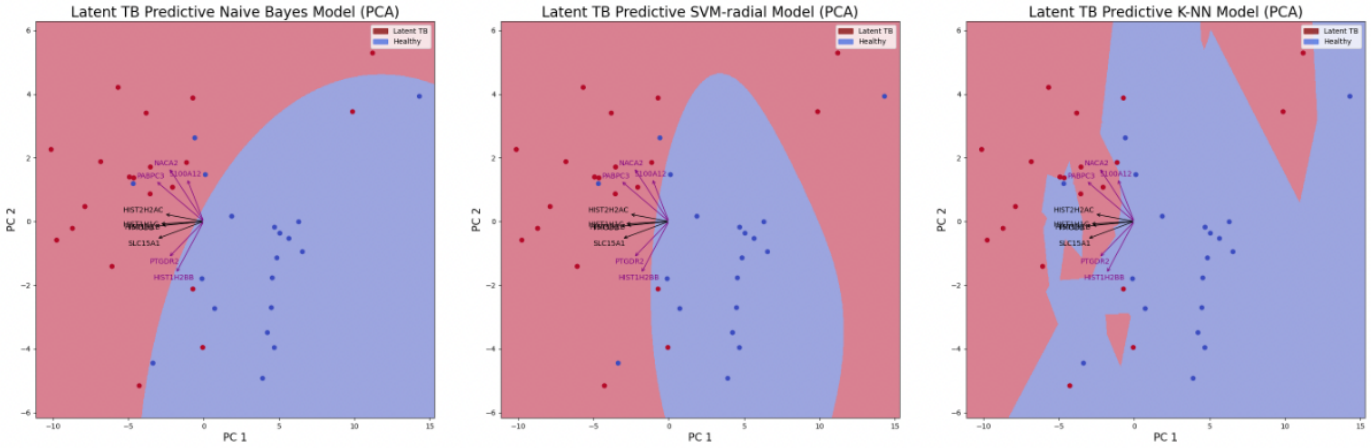


Figure 4: PCA Plots with Model Decision Boundaries and Top Associated Genes

Over a plot of PC 1 vs. PC 2, are the top associated genes with each principal component (PC1 in black arrows, and PC2 in purple arrows) (**Fig 4**). The decision boundaries for each model trained on the PC-transformed values is overlaid to provide more information on why SVMs outperform the other classifiers. Samples in the blue spaces should be labeled Healthy, whereas those in red should be labeled with LTBI. All models show some misclassification, but the NB model has the least number. KNN shows the most complex decision boundaries.

The genes that contribute most to PC1 are ***SLC15A1, TMOD2, and HIST1H1E, HIST1H1C, HIST2H2AC***. For PC2 the largest contributing genes are ***PTGDR2, PABPC3, S100A12, NACA2, HIST1H2BB***.

Discussion

Only 1 pathway was found significantly enriched using ShinyGO (**Table 1**). Although this pathway is mainly associated with the condition of Acute Myeloid Leukemia, this could imply some crossover in similar genes for Latent Tuberculosis. Potentially testing drugs that have been developed to treat Acute Myeloid Leukemia targeting this pathway could have some therapeutic power for treating latent TB. However, as aforementioned it could also be due to a strict filtering selection process. Using a Log2 Fold Change of 0.5, instead of 1, could have led to more insightful enrichment analysis. The identified top genes also provide an valuable insight for future study – specifically regarding the large number of histone coding genes.

Comparing the three different model's predictive abilities given the samples, it is apparent that SVM performed the best (**Fig 3**). Looking individually at cross-validation results, we see that it has the least tendency to over-train since the training and test scores were similar (**S.Fig 1**). K-NN with 2 neighbors performed the worst, and showed the most tendency to over-train. Using the PCA plot to observe the decision boundaries, it becomes evident why SVM did the best: Naive Bayes is a linear model and the simple delineation actually works well for most sample points but SVM's radial kernel adds just a bit more complexity to the decision boundary but doesn't actually 'trust' its training data as strictly (**Fig 4**). However, in the context of predicting new cases, this helped SVM since it was likely picking up more of the 'true' signal among the genes by focusing on the support vector genes. KNN, despite having the most complexity in terms of decision boundary, actually performed the worst for the same reason as Naive Bayes, clear by its overly complex decision boundaries. KNN was not picking up a signal among the data values, and just relying on neighbors threw off many TB samples that were close to Healthy ones.

Ultimately from this small-scale study, it is observed that SVM models may be the most fruitful in being a predictive model for latent TB. The 96 DEGs also performed strongly at differentiating samples by their log2 expressions, with an accuracy score of 84%. Moreover, heuristic models such as KNN may not be appropriate for latent TB predictions.

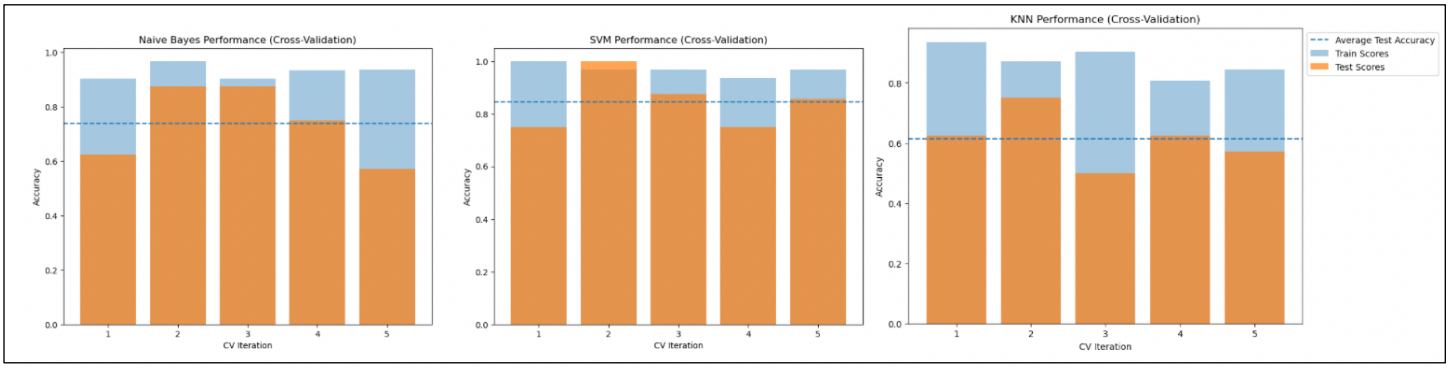
There are many limitations to this study, and these findings would be best treated as a starting point to build a stronger model by expanding the sample size beyond 39. Due to the 'wide' nature of the data being used to train (columns outnumber rows), NB, SVM, and KNN were chosen. It should also be noted that the **Fig 4** plots were trained on PC-transformed count values instead of the Log2 values used in **Fig 3**. This minimally impacted the test accuracies, and SVM still outperformed both NB and KNN. The list of 96 differentially expressed genes, also can be used to reduce the computational load of training models and reduce the 'curse of dimensionality'. The models compared were hyperparameter trained minimally, but mostly kept at default priors, providing as another next step to improve predictive ability.

References

The data discussed in this publication have been deposited in NCBI's Gene Expression Omnibus (Burel et al., 2018) and are accessible through GEO Series accession number GSE99373 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE99373)

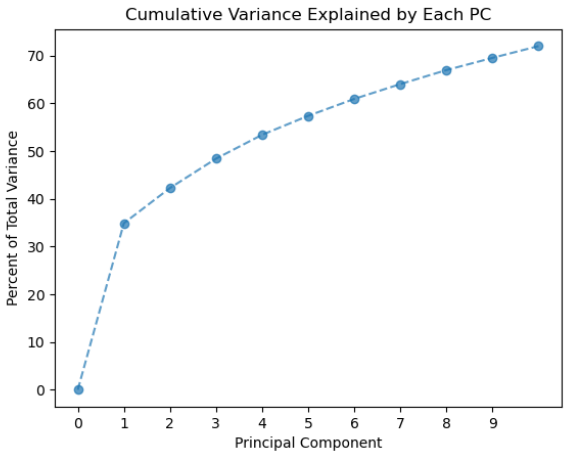
1. Burel JG, Lindestam Arlehamn CS, Khan N, Seumois G et al. Transcriptomic Analysis of CD4⁺ T Cells Reveals Novel Immune Signatures of Latent Tuberculosis. *J Immunol* 2018 May 1;200(9):3283-3290. PMID: 29602771.
2. Agathis NT, Bhavaraju R, Shah V, Chen L, Haley CA, Goswami ND, Patrawalla A. Challenges in LTBI care in the United States identified using a nationwide TB medical consultation database. Public Health Action. 2021 Sep 21;11(3):162-166. doi: 10.5588/pha.21.0026. PMID: 34567993; PMCID: PMC8455022.
3. Love, M. I., Huber, W., & Anders, S. (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." Genome biology, 15(12), 1-21.https://genomebiology.biomedcentral.com/articles/10.1186/s13059-014-0550-8
4. Steven Xijin Ge and others, ShinyGO: a graphical gene-set enrichment tool for animals and plants, *Bioinformatics*, Volume 36, Issue 8, April 2020, Pages 2628–2629, https://doi.org/10.1093/bioinformatics/btz931

Supplementary Figures/Appendix



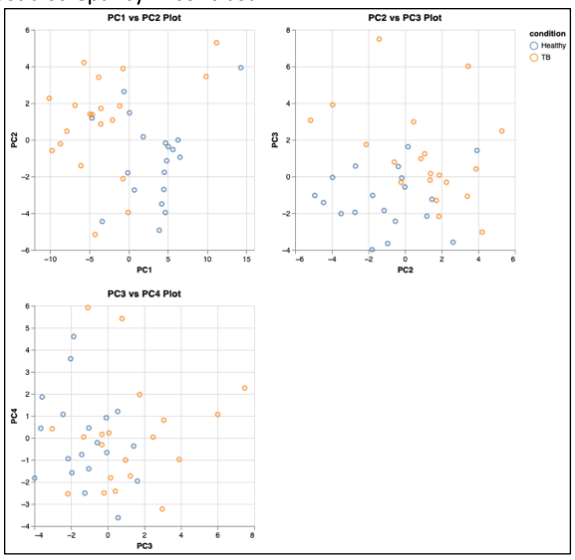
Supplementary Figure 1: Model Cross Validation Results

Figure above depicts each model's cross validation iteration, comparing training and testing scores. The mean test accuracy is presented as a dashed blue line. Figure shows that KNN faced the most tendency to over-train as apparent by large discrepancy between training accuracy (blue) and testing accuracy (orange). SVM had the least discrepancy in contrast.



Supplementary Figure 2A: Variance Explained by First 10 Principal Components

Figure above depicts the cumulative variance explained the first 10 PCs. Over 50% of the total variance is explained within the first 4 PCs. PC1 captures 34.9% of total variance, and PC2 captures another 7.37% (which are used in modelling and visualization purposes).



Supplementary Figure 2B: Comparative PC plots

Figure above depicts comparative plots of PC 1 through 4 plotting against each other to determine best clustering of sample conditions. PC 1 vs Pc 2 showed most divergence between Healthy and LTBI samples and thus was used for model visualization purposes.