Project I: Predict HIV Progression

PH 244 Big Data: A Public Health Perspective

Sarah Johnson & Rachael Phillips

February 12, 2018

Summary

Through this project, we learned about the intricacies of analyzing DNA sequences. We learned how to align sequences, create matrices to compare them, and translate DNA sequences to amino acid sequences. By utilizing `SuperLearner`'s cross-validation capabilities and by including provided baseline viral load and CD4 counts [1], we were able to surpass the prediction accuracy of Kaggle's Predict HIV Progression contest.

1 Questions

1.1 Contest Question

The Predict HIV Progression Kaggle Contest [2] requires competitors to predict the likelihood that an HIV patient's infection will become less severe, given a small dataset and limited clinical information. The contest focuses on using the nucleotide sequence of the Reverse Transcriptase (RT) and Protease (PR) to predict the patient's short-term progression.

1.2 Additional Question

We propose an additional yet similar question. Can we predict the patient's short-term progression with the *peptide sequences* of Reverse Transcriptase (RT) and Protease (PR)? Our motivation for this question is based on the central dogma of biology. Specifically, even if two individuals have different DNA sequences for a gene, they can have the same protein sequences; and since only the protein is exposed to functional constraints, then we think it will be more interesting to see the differences in the protein sequences. It should be noted that the idea to utilize the peptide sequences for prediction was suggested in the Kaggle Discussion Board.

2 Data Processing

The short-term progression status is a binary random variable (`1`: patient improved, `0`: otherwise). We do not know the short-term progression status for the test data set, so we only used the training data set (n=1000) for the analysis. Of these, 206 subjects showed improvement and 794 subjects did not show improvement. We used `R`'s `seqinr::translate` function [3] to translate the DNA sequences into amino acid (AA) sequences for our second question. We then used Bioconductor's `R` packages `muscle` [4] and `msa` [5] to align the DNA and AA sequences, respectively, since we would like to make comparisons at specific sites, and since we are interested in insertion and deletion mutations as well as simple substitution mutations. Both forms of multiple sequence alignment create `MultipleAlignment` objects. Thus, we obtained four `MultipleAlignment` objects from the two DNA and two AA sequences and we will denote them as $ma_{RT.DNA}$, $ma_{PR.DNA}$, $ma_{RT.AA}$, $ma_{PR.AA}$.

3 Data Analysis

3.1 Winning Methods

The winner, Chris Raimondi, described his methods as follows. [6] Raimondi first balanced the training set into "Yellow" and "Red" groups using `R`'s `matchControls` function to match certain features of interest including `VL.t0` and `CD4.t0`. Certain imbalances were only resolved by excluding the first 230 rows of the test data. With the training set balanced, Raimondi used `R`'s `caret` package, especially the `rfe` function, to find remaining important features, the top five of `rfe`'s selected 120 being: `VL.t0`, `QIYQEPFKNLK`, `rt184`, `CD4.t0`, `rt215`. Raimondi then trained his models and made predictions using `R`'s `randomForest` and `caret` packages (the latter for "tuning and validation enhancements"). Raimondi's final submission won with between 70% and 80% accuracy.

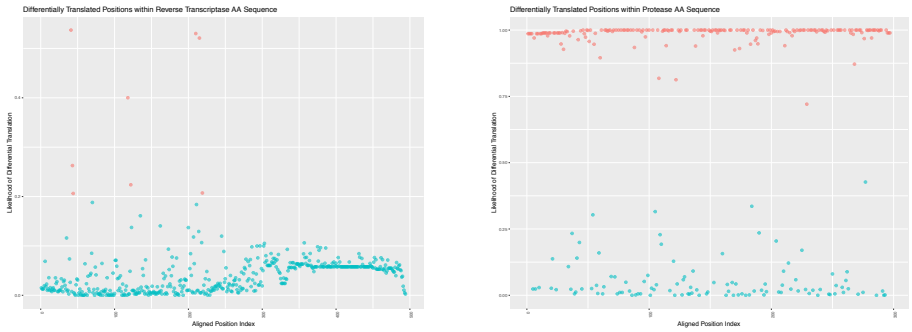
3.2 Novel Methods

3.2.1 Model Development

To identify differential DNA/AA sites, we stratified the `MultipleAlignment` [7] objects by improvement status (i.e. $ma_{RT.DNA}$ was split into $ma_{RT.DNA.1}$ and $ma_{RT.DNA.0}$). From the stratified `MultipleAlignment` objects and using Bioconductor's `Biostrings` [8] package, we created eight `consensusMatrix` (two for each sequence). A `consensusMatrix` contains the alphabet frequency for each position in the input sequences. For each DNA and AA sequence, we calculated the maximum absolute frequency difference between the two `consensusMatrix` (i.e. $|cm_{RT.DNA.1} - cm_{RT.DNA.0}|$ where cm = `consensusMatrix`). This gave us a way to compare differences across AA/DNA sequence sites between the binary improvement status groups (See Figure 1 and 2). We subsequently identified the ten sites with the highest maximum absolute frequency differences for each DNA and AA sequence. We combined the top results from the DNA sequences (using 10 sites from each gene) with baseline viral load (`VL.t0`), baseline CD4 count (`CD4.t0`), and our binary outcome `Resp` to form our model that was aimed at answering the Kaggle Contest Question. We also combined the top ten results from the AA sequences (using 10 sites from each peptide sequence) with baseline viral load (`VL.t0`), baseline CD4 count (`CD4.t0`), and our binary outcome `Resp` to form a model to aimed at answering our Additional Question. Thus, for both the Kaggle Contest Question and Additional Question, we used 22 covariates to aid in the prediction of our binary outcome.

(a) Reverse Transcriptase (b) Protease

Figure 1: Differential Expression of DNA Sequences



(a) Reverse Transcriptase (b) Protease

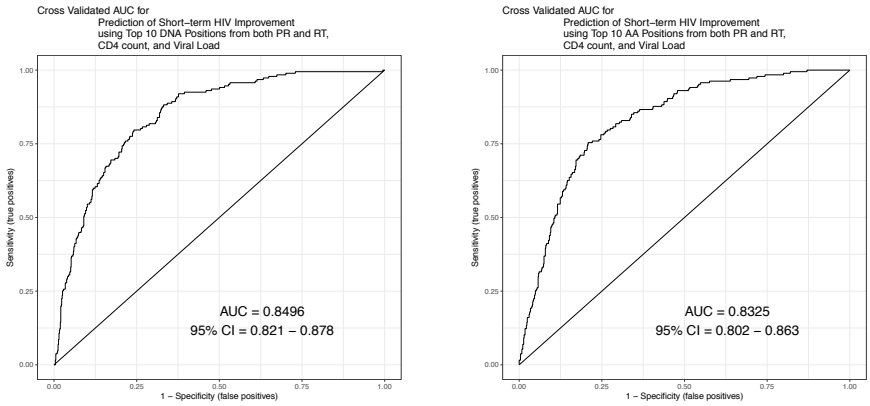
Figure 2: Differential Expression of AA Sequences

3.2.2 Prediction

For each of the two models generated, we used 10-fold cross-validation with `SuperLearner` [9] to make our predictions with the 22 covariates. SuperLearner is an ensemble machine learning algorithm that uses cross-validation to estimate the performance of multiple machine learning models. It creates an optimal weighted average of those models, using the test data performance and a specified loss function. This approach has been proven to be asymptotically as accurate as the best possible prediction algorithm that is tested [10]. Our `SuperLearner` library only contained one algorithm, `SL.ranger` (a faster implementation of random forest [11]), since this was the only algorithm we could get to converge. Thus, we did not really utilize the full capabilities of `SuperLearner`. Our specified loss function was the rank-loss (or AUC-maximizing).

3.2.3 Predictive Performance

We evaluated our predictive performance with the area under the ROC curve (AUC) - a summary measure of the predictive accuracy of a binary classification model. For the model that answered the Kaggle Competition Question (considering the top 10 Protease DNA sites, top 10 Reverse Transcriptase DNA sites, `VL.t0`, and `CD4.t0`), we achieved 84.96% accuracy in our prediction (See Figure 3), thereby outperforming the competition winner. For the model that answered our Additional Question (considering the top 10 Protease AA sites, top 10 Reverse Transcriptase AA sites, `VL.t0`, and `CD4.t0`), we achieved 83.25% accuracy in our prediction (See Figure 3).



(a) Kaggle Contest Question (b) Additional Question

Figure 3: Prediction Accuracy

4 Discussion

We would like to further explore the patterns identified within the sequences. For example, there appears to be an entire region of differentially expressed sites (as shown in red) for the Reverse Transcriptase DNA Sequence. There also appears to be quite a few differential sites within the Protease AA sequence. We hoped to see better predictive accuracy in the model that considered the peptide sequence sites opposed to the DNA sequence sites. It would be interesting to see how the predictive performance changes when the results from the Reverse Transcriptase DNA sequence are combined with the results from the Protease AA sequence. It would also be helpful to delve into the issues we were having with algorithm convergence in `SuperLearner`. Perhaps there is a better way to format the model.

References

[1] Rachael Phillips and Sarah Johnson. Big Data Project I: Predict HIV Progression. `https://github.com/rachaelvphillips/ph244-big_data`, 2018.

[2] Predict HIV Progression. `https://www.kaggle.com/c/hivprogression`, 2010.

[3] D. Charif and J.R. Lobry. *SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis.* Biological and Medical Physics, Biomedical Engineering. Springer Verlag, New York, 2007. ISBN : 978-3-540-35305-8.

[4] Robert C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, 32:1792–1797, 2004.

[5] Ulrich Bodenhofer, Enrico Bonatesta, Christoph Horejs-Kainrath, and Sepp Hochreiter. msa: an R package for multiple sequence alignment. *Bioinformatics*, 31(24):3997–3999, 2015.

[6] Chris Raimondi. How I won the Predict HIV Progression data mining competition. `http://blog.kaggle.com/2010/08/09/how-i-won-the-hiv-progression-prediction-data-mining-competition/`.

[7] Marc Carlson. MultipleAlignment Objects. `https://www.bioconductor.org/packages/3.7/bioc/vignettes/Biostrings/inst/doc/MultipleAlignments.pdf`.

[8] H. Pagès, P. Aboyoun, R. Gentleman, and S. DebRoy. *Biostrings: String objects representing biological sequences, and matching algorithms*, 2016. R package version 2.42.1.

[9] Eric Polley, Erin LeDell, Chris Kennedy, and Mark van der Laan. *SuperLearner: Super Learner Prediction*, 2017. R package version 2.0-22.

[10] Eric C Polley and Mark J Van der Laan. Super learner in prediction. 2010.

[11] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.