Predicting Live and Dead Yeast Cells with Random Forest Classification

Ryan Mahtab

August 2020

1 Objective

Our goal is to develop a method of predicting whether yeast cells are live or dead using time series data from flow cytometry readings. We want to be able to make these predictions on data from new experiments involving different techniques of killing yeast cells. The primary challenge of our task is that ground truth data for the purpose of validating our predictions is not available for most of the experiments. Our random forest prediction model will be trained on the select few experiments that do have reliable ground truth labels, however, any new data we want to classify as live or dead must be verified using a different method.

To ensure our model is optimized to predict on a specific new experiment, we build a training set from the available experiments containing ground truths that is "nearest" in distance to the new experiment. This distance metric takes into account the distributions of the flow cytometry data and will be explained in more detail in the *Method* section. After the model outputs the live and dead labels, we attempt to validate the predictions using a different set of predictions from another technique involving the optical density of the yeast cells, which will also be explained in the next section.

2 Method

2.1 Model Set Up

Our prediction model for labeling cells as live or dead is a random forest classifier model. The attributes of each cell that are used by the classifier are various channels from the flow cytometry readings, specifically the 'FSC-A', 'SSC-A', 'BL1-A', 'FSC-W', 'FSC-H', 'SSC-W', and 'SSC-H' channels. FSC and SSC refer to the forward scatter and side scatter channels respectively, while BL1 refers to the blue laser channel. These are the seven variables used by the random forest classifier to make predictions. Call this set of variables C.

When setting up the model for training, we want the observations in the training set to be similar or "near" to the observations in the prediction set with respect to these variables. Thus, when we are given a new experiment to predict on and have to define a training set with data from the experiments that have ground truth labels, we want to construct the training set so that it is "near" in regards to the channel distributions. To quantify this distance between distributions over the seven channels, we use a metric called the Wasserstein distance. If we imagine two distributions as two different piles of dirt, the Wasserstein distance between the two distributions can be thought of as the cost to turn one pile into the other, computed by the amount of dirt that needs to be moved times the average distance it has to be moved. Say the function $W(A, B)$ computes the Wasserstein distance between distributions A and B.

With respect to our objective of classifying yeast cells, each observation in an experiment has seven channel variables, each with its own distribution within that experiment over all the observations. Let us call each experiment containing ground truth labels that can be used during training a control set. Say we want to pick the 10 "nearest" of these control sets to construct our training set. We first group our prediction set by the variables 'strain_name', 'timepoint', 'inducer_concentration', and 'replicate' into what we will refer to as samples. To pick the closest control sets, we iterate over each set and calculate its distance to the samples in the prediction set. Then we aggregate the distances across samples to get the overall distance of a control set to the prediction set. Before calculating distance though, we split the current control set into its live and dead portions. This is because we want the distance of a sample to the *nearest* control. If the sample mostly contains live cells, its distribution across the seven channels should be *nearer* to the live portion of the control set rather than simply the entire control set. We want our chosen control sets to be able to distinguish live and dead after all, not just mimic the overall distribution of the prediction set.

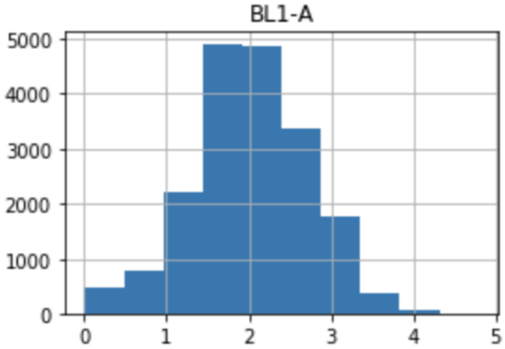

Figure 1: Distribution of the BL1-A channel of a single control set

Once we split the current control set into its live and dead parts, say L and D, we can compare each part to the current sample, say S. We then iterate over the channels and compute the distances between each part to the sample distribution, applying a log-10 transformation: compute $W(log_{10}(L_c), log_{10}(S_c))$ and then $W(log_{10}(D_c), log_{10}(S_c))$ $\forall c \in C$. We aggregate the distances over all seven channels and finally add the lower of the two sums to the overall Wasserstein distance of that control set. As mentioned earlier, this is repeated over all the samples in the prediction set and with all of the available control sets. The result is a list of control set - Wasserstein distance pairs which we can sort and then lastly return the 10 sets with the lowest values.

2.2 Model Validation

Once we have our training set defined, we can input it along with the prediction set into the random forest model and get back the predictions as output. To validate the accuracy of these predictions, we use a separate technique involving the optical density measure of each observation. The optical density analysis tells us whether a certain well in an experiment is growing or not. It is a single metric over all of the timepoints in an experiment, so to use it we must aggregate our random forest output over the 'timepoint' variable (aggregating over 'experiment_id' and 'well_id' achieves the same result). Once aggregated, we expect both datasets to have the same number of observations.

From the optical density analysis data, we treat the 'dead' column as its prediction of whether an observation is live or dead: dead = 0 being live and dead = 1 being dead. Our random forest model output returns a probability in the range $[0, 1]$, close to 1 meaning predicted to be live and close to 0 meaning predicted to be dead. So to compare the two variables, we must invert the 'dead' variable from the optical density data. After doing so, we expect observations with random forest predictions near 0 to have an optical density prediction of 0, and vice versa.

The two metrics we use to compare these two quantities are mean accuracy and mean loss. Mean accuracy is simply the percentage of observations that have the same random forest prediction and optical density prediction. Mean loss is the mean absolute difference in magnitude between the random forest and optical density predictions.

To statistically verify the relationship between these two predictions, a logistic regression model is fitted to the data. We use the set of random forest output as the predictor variables and the set of optical density predictions as the response variables. If there is a relationship between the two techniques, we should expect to see a logistic curve with a significant p-value such as the one below in Figure 2.

3 Results

With both our random forest classifier and validation technique established, we can move on to testing the model on new prediction sets. The data from the following four experiments will be used as the new prediction sets:

- YeastSTATES-OR-Gate-CRISPR-Dose-Response (20200625204022)

- YeastSTATES-CRISPR-Short-Duration-Time-Series-35C (20200625205807)
- YeastSTATES-CRISPR-Short-Duration-Time-Series-20191208 (20200610192131)
- CEN-PK-Inducible-CRISPR-4-Day-Obstacle-Course (20200721164700)

After running these four prediction sets through our classifier and computing the validation metrics, we get the following results:

Prediction Dataset	Mean Loss	Mean Accuracy	p-Value
YeastSTATES-OR-Gate-CRISPR-Dose-Response	0.1174	0.9138	0.002
YeastSTATES-CRISPR-Short-Duration-Time-Series-35C	0.0323	1	-
YeastSTATES-CRISPR-Short-Duration-Time-Series-20191208	0.2102	0.8225	0.467
CEN-PK-Inducible-CRISPR-4-Day-Obstacle-Course	0.0152	1	-

All four prediction sets yield relatively high mean accuracies, with two sets matching completely with 100% accuracy. However, the p-values of the fitted logistic regression for those two sets cannot be computed because all of the observations were of in one label group, specifically they were all predicted to be live. This causes a perfect separation error when trying to fit a logistic regression model. The experiment with 0.002 p-value results in a logistic curve shown in the figure below:

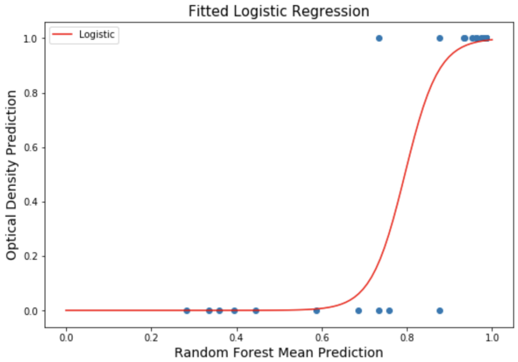

Figure 2: Scatter plot of random forest predictions vs optical density predictions with the fitted logistic regression curve

This curve helps visualize what the mean loss is telling us. The random forest and optical density predictions mostly agree with the separation of live and dead labels falling at roughly the 0.8 mark on the random forest probability scale.

The third experiment has the lowest mean accuracy and a very high p-value, which tells us that its observations were not able to be separated as neatly as the other experiments. This may indicate an issue with the relationship between the predictions of the random forest model and the optical density technique, or may be the result of an issue with the experiment itself. The results of that experiment are still being investigated.