Computer vision classification of mesenchymal and proneural glioblastoma cell lines using OpenCV and Scikit-Learn

Ryan Tsang, Chiara Mooney

March 2019

1 Introduction

As a worldwide killer, cancer has made it imperative for scientists and researchers to investigate its underlying mechanisms in an attempt to better treat patients affected with this devastating disease. A major inhibitor to our understanding of cancer is its largely unpredictable behavior. The unpredictability of cancer growth manifests itself in many forms such as variation in tumor cell population, proliferation rate, and metastatic ability. A point of particular interest is that cancer tumor cell populations are not homogeneous. Various cell lines within a tumor exhibit different properties, such as differences in responses to drugs, radiation, roles in proliferation, metastasis, and potentially more [1]. Furthermore, treatment can be optimized to given information on the tumor cell composition. In our experiments, we explore a small subset of this problem space, attempting to prove the feasibility of computer vision based classification of early stage glioblastoma (GBM) cell images as mesenchymal or proneural. While traditional laboratory methods such as immunostaining may offer insight into tumor cell composition, a computer vision solution offers immediate results at a lower cost. In general, exploring the differentiation of glioblastoma has significant relevance, as it may lead to a better understanding of the development paths of cancer cells.

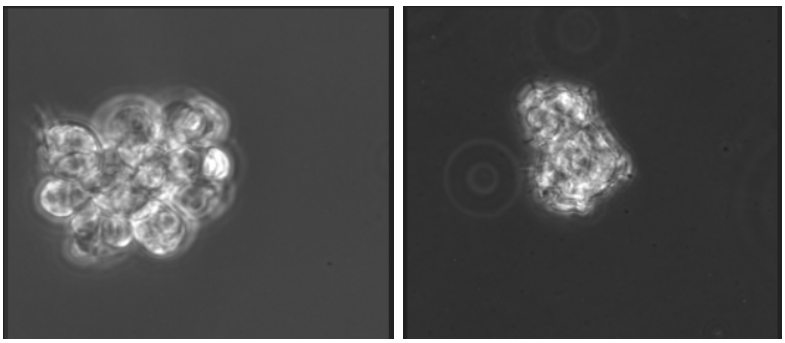

Figure 1: Example of original microscope images obtained from Seidlits lab picturing mesenchymal (left) and proneural (right) cells prior to background normalization procedure.

2 Problem

Because we are attempting to classify images as one of two labels, we naturally identify this problem as a binary classification problem. Furthermore, because we are working with images, we chose to treat the individual grayscale pixel values as inputs into our predictive model. Thus, the size of the images that we present to the model determines the number of input variables. Because we are working with mesenchymal and proneural cell lines, we assign them labels of 0 and 1 respectively. Note that we omit the possibility of images with no cells. While building the dataset for our model and experiments, we take care to filter the data to strictly images of mesenchymal and proneural GBM cells. Our goal will be to build a model that reads in the multivariate input of the images, and outputs an accurate prediction of which cell line the image represents: 0 or 1 for mesenchymal and proneural respectively. To be useful in a clinical setting, this model likely needs to perform with a minimum of 90% accuracy. However, for this project, we will claim our model is successful in classifying early stage GBM cancer cells if it has a prediction accuracy better than 50%, the random prediction case.

3 Methods

The wet lab component of this experiment was conducted by Ph.D student Alireza Sohrabi from Dr. Stephanie Seidlits' laboratory at UCLA. Two GBM

cell lines were used for this experiment: the HK280 mesenchymal cell line, and the HK301 proneural cell line. These cancer cells were seeded and cultured as 3-dimensional cell spheres. Phase images were captured at daily time points over the course of 9 days. For our computer vision classification model, we primarily used the images from days 0 through 3. These images were the ones most difficult to label with the human eye, as they have not had the chance to migrate and display distinguishing characteristics.

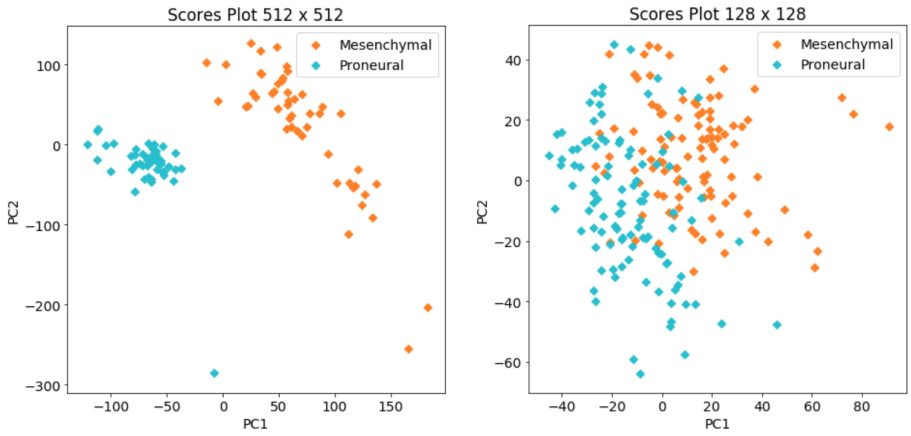

Figure 2: Scores plots for principal components 1 and 2. Clear clusters in the 512 x 512 case (left) indicate that the model discovered distinguishing factors amongst our two kinds of cells. Analysis of our image partitions revealed this factor was in fact a difference in cell sphere size. The 128 x 128 case (right) shows more overlap; this will be a more accurate representation of our model's ability to discern between classes.

The image preprocessing components of this experiment were conducted using Python and OpenCV. Before attempting to classify images, we first addressed the possibility of variation within the data set due to external factors, such as different lighting, reflections from different well plate locations, microscope models and settings, and initial cell sphere seeding size. We attempted to mitigate these factors in two steps. To normalize lighting conditions, a background subtraction was applied on all inputs. To normalize for various cell sphere sizes, we partitioned the 2048 x 2048 pixel images into smaller sizes. For our tests, we studied both 128 x 128 and 512 x 512 image partitions. With smaller image sizes, the likelihood of significant differences

in cell sphere sizes decreases. At this point, we filtered out images with no cells. For the 512 x 512 case, we obtained 50 images for each cell type, and for the 128 x 128 case, we obtained 105 images for each cell type.

We convert each image from its 2-dimensional form into a flattened 1-dimensional array in row-major order. It is useful to note that in doing so, we lose some pixel positional information which would otherwise have been preserved in a neural network. We label each image with an additional column indicating 0 for mesenchymal and 1 for proneural.

We ultimately decided to build a binary classification model using partial least squares discriminant analysis (PLSDA). There were a number of reasons why this particular choice was made. Because we are working with image data where each pixel represents a unique variable, we knew there would be many inputs to the model. More specifically, our model would have far more inputs than observations, requiring some form of regularization. Furthermore, because we are working biological data, it is highly likely that there are correlations among the input variables. These aforementioned reasons motivated the usage of a partial least squares model because it is a regularization method that captures and maximizes the covariance between inputs and outputs. Finally, because we are trying to predict classes, a classification regression method, PLSDA, makes the most sense.

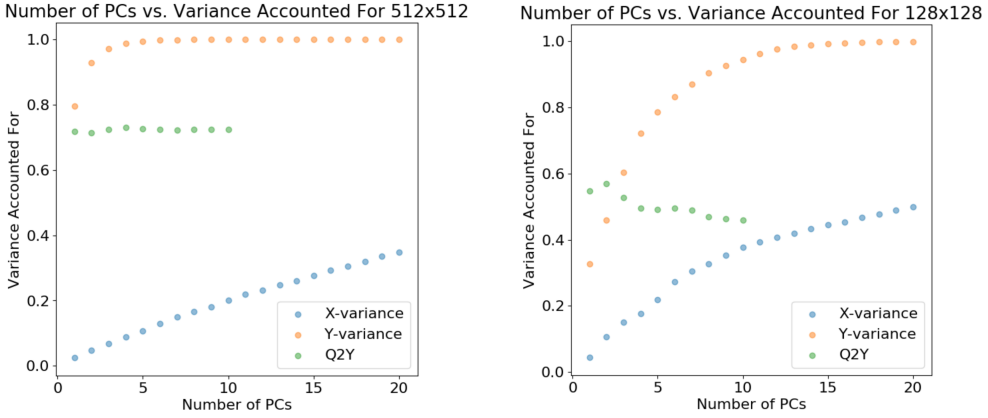

Figure 3: Plotted R2X and R2Y calculations for 512 x 512 (left) and 128 x 128 (right) cases. While 90% of the variance in y can be captured using 2 and 8 PCs, respectively, less that 40% of the variance in x is captured for both cases.

For this project, we used Scikit-Learn's PLSRegression along with a custom

thresholding algorithm to sort the results into one of the two output classes. To determine the number of components to use for PLSDA, we studied Q2Y, observing when performance of the cross validated model decreases. For both cases, we identified the optimal number of principal components to be 2. We then performed 8-fold cross validation to identify prediction error. Stratified K-fold cross validation is not needed in our case since our csv file containing the image data was constructed by concatenating the block of mesenchymal images with the block of proneural images. Thus, to separate fairly into 8 groups, we simply iterate through the observations, hashing each image to a group via observationIndex % 8. Upon completion, we analyzed the efficacy of the model via a receiver operating characteristic (ROC) curve and it's area under the curve (AUC). For our ROC curve, our true positive rate represents the number of proneural cells accurately predicted divided by the number of proneural cells. Our false positive rate represents the number of proneural cells inaccurately predicted divided by the number of mesenchymal cells. Since every cell which is not predicted as proneural will be predicted as mesenchymal, displaying the ROC for proneural cells will be sufficient in displaying all thresholding results.

We have two members on our team, the work was divided into two parts: (1) Ryan worked on building a data set from our current bank of images and carrying out calculations to analyze the accuracy of our model at categorizing cell images; (2) Chiara performed calculations to select model parameters, implemented our plsda model, and set up cross validation procedures.

4 Results

Comparing the results for the two image sizes, we note a drastic difference. The AUC of the ROC curve for the 512 x 512 case with 8-fold cross validation appears to predict with 93% accuracy. However, upon further analysis of the 512 x 512 images in our dataset, it became apparent that there were cell sphere size cues that the model may have been using. The proneural cell spheres were much smaller than their mesenchymal counterparts. In actuality, this result was the motivation to cut the original image into even smaller partitions. Upon analysis of the 128 x 128 case's ROC curve following 8-fold cross validation, we see that the AUC is a more believable 77%. The two image classes were verified to be more similar and far less distinguishable to the human eye than the 512 x 512 case.

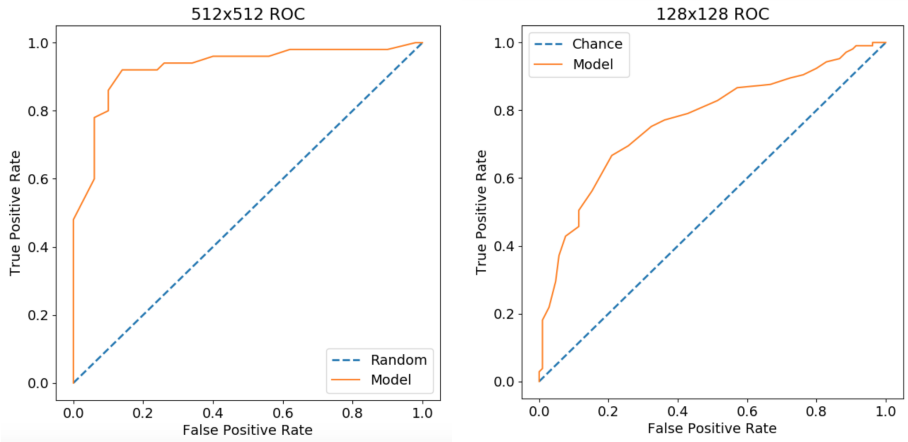

Figure 4: ROC curve for 512 x 512 (left) and 128 x 128 (right). The AUC calcuations for the curves are 93% and 77%, respectively.

It appears that the model we constructed has some predictive capabilities, keying into image data not necessarily perceived by the human eye. With a prediction accuracy of 76% in our 128 x 128 case, our model surpassed the prediction ability of the random case, 50%. In this regard, our approach was successful. However, to be feasible in any clinical setting, this result is far from desirable for a number of reasons. Firstly, the current classification success rate did not meet our proposed benchmark of 90%. Furthermore, the general problem we wish to address - classification of tumor cells into many categories - is a more complex classification problem with far more than 2 categories, increasing the chance of misclassification. It may be possible to increase prediction accuracy by including some easily attainable non-image data to supplement the inputs. However, this trade-off will likely result in increased costs and time-to-diagnosis. Unfortunately, after optimizing for the number of principal components used, we exhausted much of the tunable parameters for PLSDA, proving that it is likely beneficial to move towards less interpretable but more effective models such as neural networks.

References

[1] Sharma, S. V, Haber, D. A., Settleman, J. (2010). *Cell line-based platforms to evaluate the therapeutic efficacy of candidate anticancer agents.* Nature Reviews Cancer, 10, 241. Retrieved from https://doi.org/10.1038/nrc2820