Machine Learing Identification of Triple Negative Breast Cancers using mRNA Profile as Features with Reduced Potential of Leakage

David Chen, Ph.D.

May 27, 2021

1 Background

Triple Negative Breast Cancer is the most aggressive breast cancer subtype. Such cancers limited treatment options because they do not express ER, PR, and HER2 hormone receptors that can be targeted by endocrine drugs. Early detection of the "triple-negative" status is critical. Determining whether a patient's cancer has the triple-negative status would require effort from a clinician. The process can require time and cause delay. Further and more importantly, the assessments by clinicians might not be consistent.

Machine learning and AI have been used for more accurate cancer diagnosis. Specifically, machine learning can help distinguish cancer from healthy,, non-caner cells with better accuracy and consistency. Similarly, machine learning can also be used to further stratify cancer subtypes. The input for such classifiers can be very flexible, that is, the features can be derived from clinical information, images, or molecular/genomic data.

Related works (refs.[1] and [2]) have attempted to address the breast cancer subtyping problems using machine learning. Ref.[1] uses a regression approach to individually estimate ER, PR, and HER2 receptor status, which isn't a direct approach to assign Triple Negative Breast Cancer status. Ref. [2], published within the last 2 months, achieved impressive accuracy but rather low precision and recall. It is not surprising because of the severe class imbalance: Triple Negative Breast Cancer comprise about only $10 - 15\%$ of all breast cancers.

In this project, I will use machine learning to determine whether a breast cancer sample is Triple Negative Breast Cancer or not. The input of the classifier is that patient's mRNA data ($\in \mathbb{R}^{m \times 1}, m > 20,000$). I will make use of cloud computing with fast, reliable hyperparameter tuning implemented in AWS Sagemaker.

2 Problem Statement

The *primary goal* of this project is to build a binary classifier that identifies breast cancer samples (rows) that are "triple negative" (i.e. Label=1) using mRNA data of m genes (columns).

3 Datasets and Inputs

The Cancer Genome Atlas (TCGA) is a population-scale cancer study with over $1,000$ breast cancer patients. The majority (>90%) of these patients will have mRNA data (features) and

TNBC status (class label, based on attributes measured by clinicians) [3]. The data set is publicly and freely available. For this project, the mRNA features and clinical data (used to infer class labels 1 vs. 0) were downloaded from cBioPortal.

More specifically:

- The specific TCGA dataset I will use is the TCGA Breast Cancer dataset (code TCGA-BRCA). This dataset has approximately $1,100$ patients or rows.
- The number of columns (or genes) available is approximately $20,000$. However, many genes have missing values due to issues such as technical problems with instrument measurement. I plan to exclude genes (columns) with $> 50\%$ missing values perform mean-imputation for the rest still with missingness.
- There may be concerns regarding the dimensionality of the data. This is part of the nature of the problem in hand and also why the problem is interesting and worthy of investigation. I will consider using dimensionality reduction methods (including PCA and t-SNE) for data exploration purpose and possibly feature engineering for machine learning.

4 Solution Statement

Given $n \approx 1,000$ patients, the expected solutions will be a $\mathbb{R}^{1\times n}$ array of class labels predicted by fitting the *trained machine-learning model* to the mRNA values of a given breast cancer patient.

5 Benchmark Model

A recently published work on the subject [2] showed the best model is SVM with 10-fold cross validation. This model achieved accuracy of 0.9 but F1 score of 0.67 due to severe class imbalance.

I will build this benchmark model from scratch following the authors' written descriptions using Python/SageMaker. I expect that the results to vary/differ from those of the original authors, but I will use whatever I will have determined as the new benchmark for comparison.

6 Evaluation Metrics

During model training, I will use area under the receiver operation curve (AUROC), and visually inspect the ROC curve itself. I will also inspect metrics including F1 score (see below), precision, recall, accuracy, and mean accuracy per class.

Due to the class imbalance (the positive class is very rare), accuracy is a particularly poor choice. Therefore, for actual evaluations, the primary metric will be *F1 score*:

$$F1 = HarmonicMean(precision, recall)$$

which is already implemented in Python libraries *sklearn*. Component metrics (precision and recall score) and related metrics (e.g. specificity, accuracy score per class), will also be reported.

7 Projected Design

7.1 Data Preprocessing

A key item pointed out by the Udacity reviewer is listed as the second to the last bullet point below.

- Access and download mRNA data and clinical (meta) data from the online repository, cBioPortal (`https://www.cbioportal.org/`).
- Define class label based on clinical data downloaded.
- Select patients with available data and labels, and those meeting the following criteria: gender female, tumors non-metastatic
- To prevent data leakage, genes ESR1, PGR1, and ERBB2 (HER2) will be removed from the feature space.
- No major additional feature preprocessing is necessary, since the mRNA features downloaded are already normalized as "Z-scores" and as structured data frame.

7.2 Exploratory Data Analysis

- Confirm all patients have had informed consent – for ethical reasons (even though the data is publically available)
- Perform unsupervised learning of features to identify data clusters, and color code by class label.
- Perform univariate statistical tests (e.g. Student's *t* test, Fisher's Exact Test) to determine if the class label and other available or calculated information are significantly related.

7.3 Classification and Hyperparameter Tuning

All models will be built using AWS Sagemaker, Python 3.6 with libraries/frameworks including PyTorch and SKLearn.

- Perform feature selection using L1- or model-based approaches, implemented in *scikit-learn*
- Train the benchmark modelwith *Sagemaker SKLearn*.
- Use AWS to train and optimize hyperparameters for several popular machine learning models suitable for high-dimensional data, including:
  1. **XGBoost (Sagemaker)**: maximum depth, number of boosted rounds, early stopping rounds
  2. **Custom Neural Network (Sagemaker PyTorch)**: optimizer, learning rate, drop out rate, epochs, batch size
- Compare each classifier trained with the benchmark, and with each other.

7.4 Final Evaluation

Select the best model from each method/classifier, and apply to the hold-out test set. To prevent data leakage, the hold-out test set will only be used at the very end and not exposed to any classifiers during training. Metrics and plots mentioned in the **Evaluation Metrics** sections will be reported here as well.

References

1. Brian D Lehmann, Joshua A Bauer, Xi Chen, Melinda E Sanders, A Bapsi Chakravarthy, Yu Shyr, Jennifer A Pietenpol, et al. Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of clinical investigation*, 121(7):2750–2767, 2011.

2. Jiande Wu and Chindo Hicks. Breast cancer type classification using machine learning. *Journal of Personalized Medicine*, 11(2):61, 2021.

3. Cancer Genome Atlas Network et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61, 2012.