Predicting Genetic Risk of Developing High-Grade Prostate Cancer using Somatic Single-Nucleotide Polymorphisms

Nirmal Krishnan
Johns Hopkins University
9 E 33rd Street, 308
Baltimore, MD 21218, USA
nkrishn9@jhu.edu

Emily Brahma
Johns Hopkins University
9 E 33rd Street, 508
Baltimore, MD 21218, USA
ebrahma1@jhu.edu

Abstract

Prostate cancer is a disease that affects a significant proportion of the male population, but in most cases, the disease is harmless. However, for the small portion of the population with high-grade prostate cancer, the disease can be extremely debilitating, causing painful symptoms and even death. Prior studies have been performed, showing how hereditary factors, alcohol consumption, sexual activity, family history and race can each play a role in developing high-grade PCa. We believe this is the first study that attempts to predict genetic risk of developing high-grade prostate cancer using somatic single-nucleotide polymorphisms. We used somatic SNP profiles and clinical data from the National Cancer Institute for 500 patients diagnosed with prostate cancer. Before training our data, we performed principal component analysis on the training set. Our test results showed that a neural network performed better than a support vector machine or logistic regression. However, the neural network is still not a great predictor by itself, so physicians should consider a combination of factors- somatic SNP data, hereditary factors, alcohol consumption, and others- when determining a patient's risk of developing high-grade prostate cancer.

1 Introduction

An estimated 80 percent of men who reach the age of 80 have prostate cancer (PCa) cells. This may sound alarming, but disease progression is generally slow with low mortality. For most men with low-grade prostate cancers, the best treatment is no treatment at all, as the risk of complications from surgery or radiation therapy outweighs the generally minor symptoms of the disease. However, for the small subset of men diagnosed with high-grade prostate cancer, the disease can be extremely worrisome in terms of symptoms, disease progression, metastasis, and mortality. In these cases, the most important part of a good prognosis is an early diagnosis. With an early diagnosis, a procedure like a prostatectomy, in which the full-prostate is removed, generally results in long-term disease free survival. Therefore, it is critical that these cancers are caught early so that patients can take advantage of these surgical options, rather than radiation therapy (which typically has worse complications rates and poorer long-term disease free survival) (Stangelberger et al. 2008). In this paper, we attempt to use (somatic) single-nucleotides polymorphisms to build a classification engine that can predict genetic risk of developing high-grade prostate cancer. What makes our study unique, is that this is the first study that attempts to predict high-grade prostate cancer, rather than general prostate cancer, which most men develop anyways.

2 Background

2.1 Vocabulary

Since this paper involves several genomic and biological terms, it is important to first define some vocabulary in order to understand the study.

- **Single Nucleotide Polymorphisms (SNP):** a variation in a single base pair in a DNA sequence.

- **Gleason Score:** a measure (from 2-10) of the aggression of prostate cancer cells based on clinical pathology of prostate tissue.
- **High-Grade Prostate Cancer (HGPCa):** a form of prostate cancer that is likely to metastasize and generally results in poor patient outcomes (mortality, complications, and long-term disease free survival). A gleason score from 8-10 is indicative of High-Grade prostate cancer, while scores below 8 are considered not as severe.
- **Somatic Mutations:** genetic alteration acquired by a cell that can be passed to the progeny of the mutated cell in the course of cell division. Somatic mutations are frequently caused by environmental factors, such as exposure to ultraviolet radiation or certain chemicals.

2.2 Related Work

To our knowledge, this is the first work that attempts to predict risk of developing HGPCa using somatic SNP data. In 2009, Agalliu et al found that a single SNP in the BRCA2 protein was associated with increased risk of developing high-grade prostate cancer. However, a large twin study estimated that 42% of disease variation can be attributed to hereditary factors and 58% to environmental factors. Environment factors can cause somatic mutations and Agalliu et al's study does not consider somatic mutations, so we believe our study diverges significantly from this one. Additionally, in our study, we build a classification engine to predict patients at risk of developing HGPCa, instead of a genome-wide association study, which we believe will have more prognostic value for physicians and health care providers.

Other studies have been conducted that relate lifestyle and hereditary factors to HCPCa, for example alcohol consumption, ejaculatory frequency, family history, and race (Papa et al, Papa et al, Chen et al, Na et al). These studies show that this problem is multifaceted, and we cannot expect a perfectly causal relationship between somatic SNP data and PCa grading; however, our model could be an additive tool when combined with these other factors.

More generally, there are few studies that use somatic mutations as a predictor of disease risk, so we believe this paper is a useful addition to the literature in demonstrating its potential prognostic value.

3 Data

3.1 Source

The data used in this paper is publicly available through the National Cancer Institute GDC Data Portal, and can be found online under Project TCGA-PRAD.

3.2 Properties

This data set consists of genomic information and clinical pathology reports belonging to roughly 500 patients who have been diagnosed with prostate cancer (a Gleason Score anywhere between 2 to 10). Of the genomic information in this dataset, we used the SNP profiles of each patient. These SNP profiles specifically contain patient somatic mutations along with the position of each mutation on their genomes. The clinical pathology reports were also critical for our analysis because they contain the Gleason scores for each patient, which allows us to determine whether a patient has HGPCa. With respect to how the samples were actually drawn, all patients were sequenced using the Illumina HiSeq 2000 Sequencing System shortly after they were diagnosed with prostate cancer.

4 Methods

4.1 Pre-Processing

Significant amounts of pre-processing needed to be performed in our to make the data usable in any way. First (and most challenging), we had to consolidate our data across hundreds of clinical pathology reports and SNP profiles. After consolidation, we constructed a feature matrix with 500 rows and 19,950 columns in which each row corresponds to a patient and each column corresponds to a single SNP. We encoding our SNP data in the following matter:

Homozygous	Heterozygous	Homozygous
Both Alleles Match Reference	One Allele Matches Reference	Neither Allele Matches Reference
0	1	2

Our labels were encoded in the following way: if a patient has a gleason score greater than or equal to 8, they received a label value of 1. Otherwise, patients with a gleason score from 2-7 received a label value of 0.

In order to make sure that our models are never trained on testing data, we split the data into two segments: training and testing. This step is crucial to avoiding overfitting, in which it appears a model performs very well, but will generalize extremely poorly to novel data. The training data includes a set of instances used for learning to fit the parameters of the machine learning algorithm. The testing data is a set of instances used to assess the performance of a fully-trained classifier.

We also applied the adjustments described below to each column in order to normalize our data (train and test). The mean and std represent the mean and standard deviation of a column:

$$data(row, col) = \frac{data(row, col) - mean}{std}$$

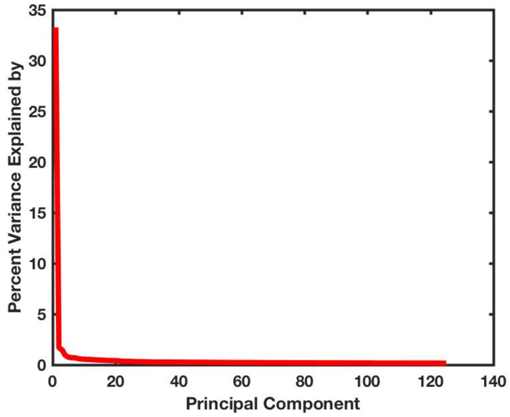
4.2 Principal Component Analysis (PCA)

An obvious issue with this data is the ”curse of dimensionality.” The data has nearly 40 times as many features as it does instances. Additionally, the data is extremely sparse, with some columns having only one non-zero entries. Therefore, initial machine learning algorithms on this large feature space resulted in extremely poor results. In order to combat this problem, we performed principal component analysis on our feature data. The table below contains the percent variance explained in the data by the first and last principal components (PC) for a variety of number of components.

Number of PCs	First PC	Last PC
1000	33.2%	0.0060%
500	33.2%	0.0060%
125	33.2%	0.1593%

From the table above, we can see that the last PC for both the 1000 and 500 parameter PCA is essentially negligible in terms of explaining the variance in the data. Therefore, we opted for the 125 parameter setting for the number of components in our PCA. The graph below shows the percent variance explained by each component for the 125 parameter setting.

Figure 1: PC vs Percent Variance Explained



Obviously, there is a steep drop in percent variance explained after the first PC. Preliminary analysis of our top two PCs with covariates (age and race) resulted in no significant confounding correlation. However, we could not investigate any correlation with batch effect or other confounders since our data lacked this information. Thus, our data could still be confounded by other undesirable covariates.

4.3 Cross Validation

In order to evaluate hyper-parameters for each of our machine learning algorithms, we used a technique called cross-validation. Cross-validation is used for the same reason that the original data set was separated into training and testing data: we do not want to train the model's hyper-parameters on data we are evaluating it on. K-fold cross-validation is a technique in which the training data is first split into K-folds, after which a unique model is trained on everything but one of the folds for all folds. The resulting cross-validation accuracies (which we will discuss more in the evaluation section below), are averaged across all k folds. We chose to do 5-fold cross validation across our machine learning algorithms.

4.4 Evaluation

We were interested in finding a model that has the maximal classification accuracy, given by the metric below:

$$accuracy = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i = y_i)$$

where $\hat{y}_i$ is the predicted label for instance i, y is the true label for instance i, and $I()$ is an indicator function, in which it returns 1 if the statement within it is true and 0 if the statement within it is false.

We used this metric to evaluate our accuracies on the cross-validation and testing sets.

5 Machine Learning Algorithms

After experimenting with numerous machine learning approaches to model our data, we landed on three techniques that we found most appropriate as described next.

5.1 Support Vector Machines (SVMs)

A support vector machines (SVM) is a classifier that finds a hyperplane that separates the two classes while also maximizing the margin between these classes. While an SVM can only find a linear separating hyperplane, we can use the kernel trick to transform a non-linear space to a linear space - allowing us to find a non-linear separating hyperplane with an SVM.

Using the Scikit-Learn library in Python, we fit our data using a linear SVM, Gaussian SVM, and a polynomial SVM of degree 2, 4 and 6. For each of these SVMs, we set C, the penalty, to one. The prediction accuracies on the cross validation sets using these fitted models are as follows:

Linear	Poly (d=2)	Poly (d=4)	Poly (d=6)	RBF
54%	62%	58%	57%	59%

From the results above, it becomes clear that the polynomial SVM with degree 2 is the best performing model. This can be explained by a couple reasons. First, when comparing the polynomial SVM with degree 2 to those with degree 4 and 6, the clear drop in performance indicates an issue with overfitting. As the degree increases, our separating hyperplane hugs the training data too closely, causing the model to perform poorly when used to predict the test data. Second, the fact that the linear SVM performs the worst can be explained by the fact that the data is not linearly separable. Thus, a linear SVM will not be able to learn the decision boundary as well as the polynomial (non-linear) or RBF SVM.

After selecting the polynomial SVM (d=2), we further fine-tuned our parameters by adjusting the penalty parameter C. The results of the accuracies from our parameter search are below:

C = 1	C = 2	C = 5	C = 10
62%	59%	55%	55%

Thus, the parameter settings that give us the best performing SVM is the polynomial kernal, with a degree of 2, and C value of 1.

5.2 Logistic Regression

Logistic regression is a technique that finds the optimal linear boundary between the two classes. In order to combat overfitting, it is common practice to add a regularization term logistic models. With logistic regression we have the option between the L1 and L2 norm. Although the only difference between these two terms is that one is simply a sum (L1) and the other is simply a square of sums (L2), these parameters can produce dramatically different results.

Using the Scikit-Learn library in Python, we fit our data using logistic regression with both an L1 and L2 penalty. The prediction accuracies on the cross validation sets using these fitted models are as follows:

L1 Penalty	L2 Penalty
59%	54%

These cross validation results tell us that logistic regression with an L1 penalty outperforms logistic regression with an L2 penalty by a noticeable amount. In general, the L1 term is expected to be very effective with sparse data, and our validation accuracy results clearly reflect that expectation. Given the significant sparsity of our data, it makes sense that the L1 regularization term leads to better results. Thus, between the logistic regression models we analyzed, the one with an L1 term is the better of the two.

5.3 Neural Network (Multi-Layer Perceptron)

A neural network (multi-layer perceptron) is a directed graph of layers of nodes - made up of an input layer, and output layer, and usually multiple hidden layers in between. Multi-layer perceptrons map a given input to a set of outputs. The goal of a neural network is to mimic the ways in which a brain functions. Thus much like the brain, to pass input from one layer to another, the input must pass a certain threshold determined by non-linear activation function. Due to its non-linear activation functions, neural networks are known to offer a complex structure capable of learning difficult data-specifically non-linearly separable data.

Using the Scikit-Learn library in Python, we fit our data using multilayer perceptrons and experimented with all of the following hidden layer structures:

- 2 Hidden Layers: 125 nodes (layer 1), 100 nodes (Layer 2)
- 4 Hidden Layers: 125 Nodes per layer

For each of these hidden layer structures, we also tried the logistic and RELU activation functions. The prediction accuracies on the cross validation sets using these fitted models are as follows:

Logistic Activation Function

2 Hidden Layers (125, 100)	4 Hidden Layers (125 Each)
58%	59%

RELU Activation Function

2 Hidden Layers (125, 100)	4 Hidden Layers (125 Each)
63%	56%

The logistic activation function did not break a 60% validation accuracy for either of the two layer configurations, indicating that it is not the right activation function for our data. When looking at the results from the RELU activation function, we can see that the configuration with two hidden layers performed much better than the configuration with four hidden layers- by 6%. This is likely because the model with four hidden layers overfit the training set - causing it to perform poorly on the validation set. Based on our validation accuracies, the best parameters for the neural network are 2 hidden layers, with 125 and 100 nodes on layer 1 and 2 respectively.

6 Results

6.1 Accuracy Metric

The accuracy metric used to evaluate the performance of our final three models is the same as explained earlier.

6.2 Accuracies for each Technique

Through the process of cross validation, we were able to pick the model with the optimal parameters from each machine learning algorithm. Running these models on the test data produced the following final accuracies:

Neural Network RELU (125, 100)	Poly SVM Degrees = 2 Margin of 1	Logistic Regression w/ L1 Regularization
59%	55%	56%

The results show that the neural network with a RELU activation function performed best, followed by logistic regression with L1 regularization and the polynomial SVM with degree 2. The neural network likely performed better due to its ability to learn complex non-linear boundaries. Despite the fact that the logistic regression model can only learn a linear boundary, it makes sense that it surpassed the SVM since the L1 regularization term allows it to perform well on sparse data.

7 Conclusions

The unfortunate conclusion of our findings is that predicting whether a patient will develop HGPCa is a difficult multifaceted problem, and none of our models are exceptional in doing so. Our results show that by itself, somatic SNP data is not a useful predictor of risk of developing HGPCa. A better model would take into account other relevant factors, like the ones discussed in the ”Related Works” section. Unfortunately, due to limitations in data resources, we could not explore this combined model of somatic SNP data with other environmental data. However, we believe a combination of these two could provide physicians with a model of high prognostic value and should be researched further.

References

Anton Stangelberger, Matthias Waldert, and Bob Djavan. 2008. *Prostate Cancer in Elderly Men*, Rev Urol, Spring.

Ilir Agalliu, Robert Gern, Suzanne Leanza, and Robert D. Burk. 2009. *Associations of High-Grade Prostate Cancer with BRCA1 and BRCA2 Founder Mutations*. Clinical Cancer Research. Vol. 15, No. 3.

Papa NP, MacInnis RJ, Jayasekara H, English DR, Bolton D, Davis I et al. 2017. *Total and beverage-specific alcohol intake and the risk of aggressive prostate cancer: a case-control study*. Prostate Cancer Prostatic. Electronic.

Papa NP, MacInnis RJ, English D, Bolton D, Davis ID, Lawrentschuk N, Millar JL et al. 2017. *Ejaculatory frequency and the risk of aggressive prostate cancer: Findings from a case-control study.* Urol Oncol. Vol. 17, No. 2.

Chen H, Liu X, Brendler CB, Ankerst DP, Leach RJ, Goodman PJ, Lucia MS, Tangen CM, Wang L, Hsu FC, Sun J, Kader AK, Isaacs WB, Helfand BT, Zheng SL, Thompson IM, Platz EA, Xu J. 2016. *Adding genetic risk score to family history identifies twice as many high-risk men for prostate cancer: Results from the prostate cancer prevention trial.* Prostate. Vol. 12, No. 12.

Na R, Ye D, Qi J, Liu F, Lin X, Helfand BT, Brendler CB, Conran C, Gong J, Wu Y, Gao X, Chen Y, Zheng SL, Mo Z, Ding Q, Sun Y, Xu J. 2016. *Race-specific genetic risk score is more accurate than nonrace-specific genetic risk score for predicting prostate cancer and high-grade diseases.* Asian J Androl. Summer.

8 Course Project Statements

Contributions from Emily Brahma:

- Processing SNP data
- Testing/running machine learning methods on data
- Wrote part of powerpoint presentation and final writeup

Contributions from Nirmal Krishnan:

- Setting up data pipeline, processing labels
- Testing/running machine learning methods on data
- Wrote part of powerpoint presentation and final writeup