Machine-Learning Approaches to Prognosticate Cancers and Elucidate Mechanisms Using Genomic Datasets

Mohamed Tageldin, Rafi Haque, Yuan He, Sapoonjyoti DuttaDuwarah, and Derek Onken

Abstract

(Tageldin, Dutta) In this report, we discuss the optimization of two well-known machine-learning approaches, K-Nearest Neighbors regression (KNN) and Artificial Neural Networks (ANN), in survival prediction and disease prognostication. We uncover the roles of 9p chromosomal deletion, CDKN2A deletion, and GATA3 protein amplification as highly-prognostic bio-markers within IDHmut-NonCodel lower-grade glioma patients. In addition, we show suggestive, but inconclusive, evidence that the IDHmut-Codel subgroup is in fact heterogeneous and separable into three subclasses. During method development, we witness that non-cumulative probability predicts survival outcomes more accurately than cumulative (Kaplan-Meier) survival functions. In high-dimensionality settings, our KNN implementation outperforms Cox proportional-hazards regression with elastic-net regularization, the current state-of-the-art in survival prediction. In addition, our novel method of local dimensionality reduction proved highly-effective in dealing with missing data, a typical problem encountered in real-world datasets.

Background and Motivation

(Tageldin, Dutta) Recent massive data-sharing efforts have revolutionized cancer research. Among the most prominent and influential of these efforts, the multi-institutional collaborative initiative known as The Cancer Genome Atlas (TCGA) provides enormous amounts of data from multiple platforms, including clinical, imaging, genomic, proteomic, and epigenomic data.[1] Commonly, number of features far outnumbers the sample size in a typical TCGA cancer cohort ($P >> N$, the famous "Curse of Dimensionality" dilemma). This abundance of data resulted in a recent shift in cancer research, where the most pressing challenges shifted from data attainment to data analysis.

Brain tumors are among the most lethal and widespread neoplasms in pediatrics and adults. The American Brain Tumor Association estimates 78,000 new brain tumor cases (primary malignant and non-malignant) in 2016.[2] Malignant brain tumors account for the most common cause of cancer-related deaths in the 15-39 age group, and are the most prevalent cancers in this subgroup. These statistics reflect the importance of pioneering research for brain tumor diagnosis and treatment, as well as the need for concentrated efforts geared towards accurate prognostication of brain tumor patients. Gliomas, the most common brain neoplasms, are subdivided into two large subsets, Lower Grade Gliomas (LGG) and Glioblastoma Multiforme (GBM), based on disease progression history, disease aggressiveness, and histo-pathological characteristics. Recent advances in cancer genomics, spearheaded by the TCGA, recently identified molecular sub-classes of LGG's with distinct survival, based on the mutation in IDH1 and IDH2 genes and the co-deletion of the p- and q-arms of chromosomes 1 and 19, respectively (1p/19q co-deletion, or "Codel").[3] Similarly, earlier efforts by the TCGA identified four molecular subtypes of GBM: Proneural, Neural, Classical, and Mesenchymal based on their genomic expression profiles.[4]

Stratification of cancer patients into distinct sub-populations, one of the pillars of modern medicine, represents a key step in developing personalized treatments. Stratification of LGG patients is possible beyond the known subtypes via epigenomic profiling, especially DNA and histone methylation.[5] In light of these findings, we report the use of genomic and proteomic machine-learning approaches to elucidate molecular pathways and mechanisms that can further subdivide and prognosticate LGG subgroups, with a special interest in the IDH-mut-Codel and IDH-mut-non-Codel subpopulations. GBM, and by association its similar subgroup, LGG

IDHwt, have already been studied extensively at the molecular level and avoid focus in this study. While we primarily focus on brain tumors, we show the performance of our algorithms on breast cancer TCGA patients for comparison.

Various challenges face survival prediction efforts using integrative approaches and genomic datasets. The high-dimensionality of the feature space makes picking important features difficult while also degrading accuracy. Additionally, rather specific to survival analysis, right-censorship presents another challenge. A patient is considered right-censored if he is lost to follow-up before the study duration ended, so the patient's label would indicate the last time he was alive rather than the time of death. To deal with these challenges, we implemented and optimized two primary survival-prediction methods, K-Nearest Neighbors and an Artificial Neural Network. We describe data handling and processing for these models and compare their performances to Cox Proportional-Hazards (PH) Regression, the current state-of-the-art and the most ubiquitous method in survival analysis.[6]

Methods and Model Development

Data Preprocessing

(Onken) The data contained many uninformative features and patients stemming from either lack of variance or lack of information. First, we normalized the features by Z-score and removed all features with zero variance. Since every patient lacked values for at least one feature and every feature lacked information for at least one patient, tossing out all non-full columns or rows would leave us with an empty matrix. Instead, we looped through "chunks" of features, removing features missing more than a tunable threshold of patients, followed by removal of patients missing more than a tunable threshold of features (**Figure 1**). Using chunks of features allows us to avoid setting one global threshold and provides better control of the feature/patient removal process. The feature chunks used, in order, are Clinical, Mutation, Gene Copy Number Variation, Chromosomal Copy Number Variation, Protein, and mRNA. This method generated the "Reduced" GBMLGG and BRCA models. This process yielded a "general" matrix with some representation of each of the GBMLGG sub-populations (referred to as "ReducedModel"). To maximize sample size within specific subsets of GBMLGG patients (particularly GBM patients), however, we tuned thresholds independently for each sub-population (**Table 1**).

Feature Selection

(Tageldin, He) We used three approaches for feature selection:

Manual method. Based on a literature search, we hand-picked the most important features in each dataset (**Table 1**). Brain cancer features were based on recent World Health Organization (WHO) guidelines and advances in the field,[3,7] while breast cancer features were mostly selected from the recent FDA-approved "MammaPrint" feature set used in breast cancer prognostication.[8]

Filter method. Filter methods—aptly named—apply some "filter" to the feature set independent of the classifier or regressor being used.[9] While known to be fast, filter methods typically exhibit poorer performance than wrapper approaches. To apply the filter method, we calculated the Spearman correlation of features with the survival of uncensored cases.

Wrapper method. Unlike filter methods, wrapper approaches rely on the performance of the classifier/regressor to provide feedback, guiding the feature selection process (hence the name).[10] We generated multiple sets of randomly-chosen features, which were used to predict survival. Features were then ranked by the average C-index of feature sets in which they appeared. The logic behind this approach is intuitive; one infers that a feature is important if it improves the C-index of the feature sets in which it was included. The typical feature appeared in 50 feature sets, and 500 total sets were used.

For the non-manual feature selection methods, features were selected using the validation set, ensuring that the testing set remained uninvolved in the selection process.

Artificial Neural Network

Data Transformation

(Haque) To prepare the data for use by the neural network, we performed two separate data transformations on the set of hand-picked features. These transformations added a time indicator as an additional input to the neural network for the purpose of predicting survival status. For the first transformation, we replicated the features for each year of survival and assigned a label indicating whether the patient was alive or dead at that time. A patient that died was not replicated after the time of failure (**Table 2**).[11] Alternatively, for the second transformation, a patient that died was replicated after the time of failure proportional to his survival time. For instance, if a patient survived 3 years, we replicated the patient's features a total of 6 times (3 with a positive label, 3 with a negative label) to create a more even distribution of classes.

This approach allows for the neural network to jointly model time with the continuous and categorical explanatory variables in the model without any outright constraints. The network outputs the estimated conditional probability of survival status as a function of the time interval. The survival time can be estimated by summing the survival status across all time points for each patient and then used to calculate the c-index.

Perceptron

(Haque) We implemented a regression $y = WX + b$ (predicted output y using features X weighted by W with bias b) using the Tensorflow software library in Python, and converted the two outputs (alive and dead) of the regression into probabilities using a softmax activation function. Using gradient descent with a learning rate of 0.1, we trained the weights of our network via a cross entropy loss function:

$$C = -\frac{1}{n}\sum_{x}\left[y \ln a + (1-y)\ln(1-a)\right],$$

where a is the predicted probability, y is the true probability, and n is the number of observations in the training data. We evaluated the performance of our model using five fold cross validation (80% training, 20% testing) using the concordance index and the area under a receiver operating characteristic (ROC) curve. Of note, adding a single or multiple hidden layers to this approach resulted in chance performance.

Artificial Neural Network

(Haque) We also implemented a neural network with two ReLU layers that yielded a single probability of survival (**Figure 2**). We used a squared error loss function and trained our weights using gradient descent with a learning rate of 0.1. We evaluated performance of our model using five fold cross validation (80% training and 20% testing) varying the number of nodes (1, 4, 7, 10, 13). Importantly, we only evaluated using the area under the ROC curve due to the inability to generate output probabilities.

K-Nearest Neighbors (KNN)

Model Selection and Assessment

(Tageldin) We used a standard train/validate/test approach for model selection and assessment, with equal assignment to the three sets. To mitigate the effect of "lucky picks," we created ten random permutations of the populations to generate ten random sample assignments to the different sets. We used the validation set to tune model parameters (such as feature selection and choosing the optimum K) and the testing set for model assessment. The C-index, the measure of the fraction of "orderable" patient pairs whose order was correctly predicted by the algorithm, acted as our metric for model performance.[12] We also reported the mean-squared error (MSE) for comparison (**Figure 3**).

Optimization: Dealing with censorship

(Tageldin) K-Nearest Neighbors has been used for predicting survival in kidney transplant patients, and the use of this prototype approach has shown advantages over Cox proportional-hazards regression when the hazard proportionality assumption was not met (using additional sets of simulated data).[13] Motivated by these findings, we investigated the performance of KNN on our dataset (**Figure 4**) using two implementations:

Kaplan-Meier Estimates (K-M). We constructed a K-M graph using the K-nearest patients and estimated the survival of the patient in question using the area under curve (AUC).[14] The K-M function is a non-parametric estimator of the cumulative probability of survival at time t of a population of patients, given

the survival time and censorship status of each patient, and is given by the equation:

$$S(t) = \prod_{t_j \leq t} \frac{n_j - d_j}{n_j},$$

where $S(t)$ is the Kaplan-Meier step function at time t which is based on n_j, the number of individuals alive just before time t_j , and d_j, the number of deaths at time t_j. We estimate the survival of the patient in question by calculating the AUC of $S(t)$ from time t_0 to time t_n:

$$AUC = \sum_{i=1}^{n} S(t_i)(t_{i-1} - t_i),$$

where $S(t_i)$ is the value of our step function just before time t_i.

Non-Cumulative Probability Estimates. We converted the outcome measure to a [0,1] binary indicator of alive/dead status at a particular time t. The probability of being alive at any time t is thus calculated as

$$P_t(Alive) = \frac{N_{Alive}}{K - N_{Unknown}},$$

where N_{Alive} is the number of neighbors alive at t, K is the number of nearest neighbors, and $N_{Unknown}$ is the number of patients who were right-censored at an earlier time point. In other words, the survival probability of the patient in question at any time t is estimated by averaging the survival status of the uncensored sub-population (within the K- nearest neighbors) at that particular time. The overall predicted survival time is thus the sum of survival probabilities at all time instances:

$$Survival = \sum_{t=1}^{t_{max}} P_t(Alive)$$

In effect, the first implementation has some "memory" of past events, as it is a cumulative probability metric. The second implementation has no memory and is simply the average survival status at each time point. The second implementation proved to be more accurate and was thus adopted for all subsequent experiments.

KNN Optimization: Dealing with missing values

(Tageldin) Missing data presents a challenge in almost all real-world datasets; our GMBLGG and BRCA TCGA cohorts were no exception. Most commonly, handling this problem involves deletion and/or imputation. Deletion may cause drastic decreases in sample size while imputation, even when performed intelligently, makes assumptions and guesses to fill-in the missing data spots. While our implementation relies on deletion (**Figure 1**), we found that, when the right assumptions are met, we can deal with massive amounts of missing data without any significant loss of prediction accuracy. This approach exploits the prototype-nature of KNN by only comparing patients along dimensions that they share. This sort of local dimensionality reduction displays its peculiarity by not depending on any inherent superiority/distribution of certain dimensions, but rather on their mutual availability for the compared patients. Of course, one might imagine a scenario where two patients have very few common dimensions, along which the patients' distance is small. Such a scenario might cause a false impression of extreme proximity between patients who might have otherwise been considered further apart (if there were not so much missing data). To mitigate this effect we normalized the number of dimensions along which the patients are being compared. The resultant distance equation thus becomes:

$$Dist = \frac{1}{\lambda dim} \sum_{i=1}^{K} |C - S_i|$$

Where C is the center subject, S_i is one of the surrounding (K- nearest) neighbors to C, dim is the number of common dimensions divided by total number of dimensions, and λ controls the penalty on lack of common dimensionality. In other words, sharing many dim leads to a low $Dist$, and a large λ implies a high penalty for lack of common features. When there are no missing values (i.e. dim = 1) and λ = 1, the distance equation behaves the same as the standard 1-norm.

Model Interpretation

(Tageldin) We used the "ensemble feature selection" method (the aforementioned wrapper method) to facilitate model interpretation. After shuffling patient samples and assigning them to training and testing sets, we ranked the features by the accuracy of ensembles in which they were included. The process was repeated

over 10 trials, and the final rank of features was obtained by calculating the median ranks over all trials. To facilitate interpretation, mRNA features (many of which have obscure or unknown roles) were excluded from the model interpretation experiments.

Cox Proportional Hazards Regression

(He) For the "basic" models containing a handful of manually-selected features, Cox regression without regularization was applied using MATLAB's `coxphfit.m` function. For the "reduced" models (with 17,000+ features), we used the `glmnet` package that applies Cox PH regression with Elastic Net regularization.[15]

Results and Discussion

A Note on Kaplan-Meier Curves

(Tageldin) Surprisingly, we observed that non-cumulative survival probability was more predictive of survival outcomes than cumulative ("Kaplan-Meier") probability functions, used ubiquitously in the biological and clinical literature (**Figure 3**). Since the K-M curve shows the probability of survival at any time t, one can predict the survival of the "typical" patient in the cohort by integrating the AUC. Now, if we compare this predicted survival with the actual survival of the "typical" patient we would have a direct method of testing our hypothesis. However, there is no actual "typical" patient, and one could not take the average survival of samples because of censorship. In our experiments, we additionally generated multiple cohorts of homogeneous patients by choosing the K most similar patients to the "central" patient, who in this case was considered the "typical" patient in the patient cohort. Now if we compare the actual survival of the "central" patient (who was not used to generate the K-M curve) with the area under the K-M curve, we would have a direct, robust method of testing our hypothesis (**Figure 4**).

While we cannot definitively determine the generality of this result due to lack of enough experiments, we believe the aforementioned experiment provides some preliminary evidence supporting such a generality, the implications of which are far over-reaching.

KNN Model Performance

(Tageldin, He) Our final KNN model relied on non-cumulative probability estimates for dealing with censorship, used the 1-norm as the distance metric and relied on a special form of local dimensionality reduction to deal with missing data. In a "typical" experiment, the optimum K ranged between 25 to 55.

One can see that the KNN model (with various feature selection methods) achieves comparable performance to Cox regression when the number of dimensions is very small and a superior performance to Cox regression with Elastic Net regularization when the number of features is very large (**Figure 5**). Also notable, KNN model performance achieves better performance on the GBMLGG dataset than the BRCA dataset. We believe this stems from the fact that the GBMLGG dataset contains a very heterogeneous set of patients, separable into distinct classes with differing characteristics and survival. This effect can be seen where the LGG population has lower prediction accuracy than GBMLGG, but higher prediction accuracy than its own constituting subpopulations (**Figure 5 (lower left)**). Another observation is that the IDHmut-Codel subpopulation has a highly-variable survival prediction accuracy, indicating that the random allocation of patients into training, validation, and testing sets has a large effect on accuracy, suggesting that this subgroup has variable characteristics and, potentially, sub-classes. To investigate this possibility, we performed a Cluster-of-Clusters (CoC) analysis, and observed distinct survival and genomic profiles of three IDH-mut-Codel sub-populations, but not for the IDH-mut-nonCodel subtype (**Figure 6**).

The small subset of features chosen from literature review clearly achieves comparable performance to the pre-processed model, with or without feature selection. This holds true for both the brain and breast cancer datasets, suggesting that either these were the most important features or there were many important, but redundant, features that are correlated with each other. We found some truth to both explanations given the results of the feature correlation experiments (described below) and the fact that many of the manually-selected features rank highly in the model interpretation experiments.

While feature selection helps, the optimized KNN algorithm appears able to handle very high dimensionality without major reduction in prediction accuracy. In fact, even with over 17,000 features, the KNN

algorithm achieves comparable performance to the basic model containing only a handful of handpicked features. Rather rare for any prediction algorithm, this *particularly* is not true for Cox PH regression, where the performance drastically worsens with high dimensionality, even when elastic net regularization is used.

Within reasonable limits, deletion of features/patients with lots of missing values is preferable to keeping the missing values to maximize sample size. Dramatic increases in C-index values of the "reduced" models in comparison to the corresponding original models demonstrates this, even though both models have a comparable number of features. Nonetheless, KNN achieving a median C-index of 65% with the original model attests to the method's ability to handle missing data.

Missing Data and Feature Correlation Experiments

(Tageldin, He) To interrogate the handling of our KNN implementation of missing values, we randomly removed values from the "reduced" brain cancer dataset (containing almost 500 patients and 17,000+ features) and checked the accuracy of survival prediction (**Figure 7, Panels A and B**). The extremely robust results were resistant to accuracy drop even when 95% of values in the feature matrix were artificially removed! We hypothesize that this robustness stems from the correlation of the features; when one feature is missing, the other features "compensate" and provide essentially the same information. To support this hypothesis, we show that: features are indeed highly correlated (**Figure 7, Panel D**), subsets of correlated features generally yield higher accuracy than uncorrelated features (within limits) (**Figure 7, Panel C**), and *random* deletion of up to 95% of features failed to reduce prediction accuracy (**Figure 7, Panel E**)! The latter result surprises us since the "common wisdom" suggests incorporating features that are as de-correlated as possible when making predictions using supervised machine-learning approaches. While the common wisdom holds true for very high feature correlations, some degree of correlation appears optimal for the KNN method.

KNN Model Interpretation

The KNN model not only emphasizes features that are excellent prognosticators in the various brain cancer subgroups (**Figure 8**), but also that the features it picks for the large populations (GBMLGG and LGG) are the same features known to correspond to their constituting subgroups.[3,4] Even within GBM, where the K-M p-values lacked significance, we observe that the curve separation is well-marked and that the non-significant p-values stem from a lack of hazard proportionality (the curves cross at around 700 days for all top features), indicating that these features are associated with better survival at the early stages, and poorer survival afterwards.

The top three features for the undivided dataset were IDH1 mutation, PTEN deletion and 10q chromosomal arm deletion, in order of importance. The top prognosticators also included, as expected, the cell cycle control genes Cyclin E1 (CCNE1, a known oncogene) and CDKN2A (a known tumor suppressor gene).[16] Counter-intuitively, the survival of IDHmut patients is known to be better than that of IDHwt, due to the production of the oncometabolite 2-HG, which causes altered DNA and histone methylation and, presumably, suppression of stem-cell phenotype.[16,17]

Within the IDH-mut-NonCodel subgroup, the three top prognosticators were 9p chromosomal arm deletion, CDKN2A deletion, and GATA3 protein amplification. Chromosomal deletions are among the most frequent features found in cancer, and are often associated with worse prognosis due to the deletion or inactivation of tumor suppressor genes.[16] 9p chromosomal deletion is known to be a poor prognosticator in GBM,[18] and we show that it is, in fact, the best prognosticator in the IDHmut-nonCodel subgroup. Peculiarly, GATA3 protein amplification is associated with *better* survival, a finding rather rare for protein amplifications. Unlike IDH mutations, however, we could not find a plausible mechanism for this phenomenon in the published literature. As a matter of fact, it was shown by Molenaar et al that GATA3 upregulates Cyclin D1 (CCND1, a known oncogene) expression in neuroblastoma cells[19]! These finding suggest that GATA-3 either plays different roles in glioblastoma and neuroblastoma, or that it has other roles that are yet unknown. The ranked feature list for each of the GBMLGG sub-populations is included in the supplementary material.

Perceptron and ANN Model Performance

(Haque) We evaluated the performance of an artificial neural network after implementing two different data transformations[11] for both the brain and breast cancer data sets. We compared the average AUC across five

folds for the GBMLGG data set as a function of the number of hidden nodes (**Figure 9 (***top, left***)**). We found that max performance for transformation 1 (AUC = 0.70 $\pm$ .09 SD, H=13) outperformed max performance for transformation 2 (AUC 0.65 $\pm$ .08 SD, H=4) in predicting survival status. The ANN, achieving chance level performance for both the transformation methods, lacked success in predicting survival status for the BRCA dataset (**Figure 10**).

We also evaluated the performance of a perceptron in predicting the survival status of the patient using both concordance index and the AUC as metrics. We found that the perceptron significantly outperformed the the ANN in both the brain (AUC 0.75 $\pm$ 0.01 SD) and breast cancer data sets (AUC 0.79 $\pm$ .01 SD) when using transformation method 2 (**Figures 9 and 10**). For the GBM data set, we also observed a C-index of 0.71 and 0.66 for transformations 1 and 2, respectively, after conversion of survival status to survival time.

According to Yang, the area under an ROC curve provides an estimate for concordance index;[11] however, when using an identical data transformation method to the previous group, we found that C-index underestimated AUC. Such differences could be attributed to attaining survival time based on survival status. Survival time, calculated by summing the survival status of all time points for a single subject, resulted in integer representations of survival time in years. These survival time representations increase the likelihood of ties when calculating C-index and, therefore, may contribute to a possible reduction in C-index when compared to AUC.

We often found that the survival time of a patient was 0 or the number of times the patient's features were replicated. Thus, the perceptron predicted the survival status of the patient independent of time indicator. If a particular feature has significance, for instance, age in the brain cancer data set, it may lead to the same prediction of survival status regardless of time indicator. This propensity for a survival time of 0 may further contribute to a greater number of ties and reduction in C-index.

To remove this issue, we could have increased the resolution of our time indicator from years to months. However, this increase in resolution comes at the cost of increasing the number of times the features are replicated, resulting in a highly correlated feature set. In combination with an imbalanced class distribution, a highly correlated feature set may have been the reason for a poor performance of the ANN on the breast cancer data set—21 features compared to 5 features for the GBMLGG data set. Given these issues, we suggest using simple logistic regression over an ANN to assess survival status rather than comparing survival time between two patients. To remove these confounds further, we suggest using methods that directly predict survival time rather than relying on data transformations that lead to highly correlated feature sets and imbalanced class distributions.

Future Directions

Further work is needed to elucidate the pathways involved in IDHmut-nonCodel prognostication using gene-enrichment analysis. This should be complemented with hypothesis-driven research to uncover the mechanisms behind GATA3 protein amplification acting as a marker of good prognosis in IDHmut-nonCodel patients. In addition, more comprehensive clustering approaches need to be undertaken to investigate the possible separability of IDH-mut-Codel patients into 2-3 classes with distinct survival and genomic profiles. On the method development side, further work needs to investigate: the accuracy of non-cumulative vs. cumulative probability in survival prediction, the effect of K-D trees indexing on the speed and accuracy of our KNN algorithm, and the comparative performance of our local dimensionality reduction approach vs. ordinary imputation in handling missing values.

Supplementary Materials

KNN Optimization: Attempts at gradient descent-based optimization

(Tageldin, Onken) We tested two implementations for tuning K. In one implementation, we used a traditional grid-search approach, where we varied the number of nearest neighbors gradually and adopted the number of nearest neighbors yielding the highest C-index on the validation set. In another implementation, we chose a large K and used a 1-D Gaussian filter to emphasize nearest neighbors in a continuous fashion, instead of the discrete grid-search approach. First, we created a Euclidean filter that reflects the relative distances of the K-nearest neighbors to each other. Then, we combined the Euclidean filter with the afore-

mentioned Gaussian filter to yield the final filter used to determine the weight of each neighbor. For this Gaussian distribution, we determined the Gaussian filter value G_i for neighbor i by

$$G_i = e^{\frac{-X^2}{2\sigma^2}},$$

where x is i's distance from the center and σ is the standard deviation of the fitted Gaussian. We later normalize G_i by the sum of all elements in the Gaussian filter vector. The purpose behind this procedure is to ensure that the weight of any neighbor is determined both by its relative Euclidean distance and its probability mass as determined by a Gaussian distribution. In order to further optimize survival prediction, we investigated the effect of emphasizing certain features over others in distance calculation. A shrinkage factor β_p was applied to each feature p in the distance equation, reflecting how de-emphasized it is in determining the nearest neighbors. Thus the modified distance equation is given by:

$$Dist = \sum_{i=1}^{K} \beta^2 |C - S_i|$$

where β is the vector containing these shrinkage factors.

We used gradient descent to find optimal values of σ (of the Gaussian filter) and β using a mean squared error cost function:

$$MSE = \frac{1}{N} \sum_{i=1}^{n} (f(x_i) - y_i)^2,$$

where we compare the output of our function f to the known associated output y for subject i of our N subjects. Since KNN is a prototype method, we lacked a differentiable function to calculate the gradient of the cost with respect to Sigma or Beta. To overcome this challenge, we approximated the gradient by perturbing σ or β and calculating the partial derivative using the equation:

$$\frac{\partial MSE}{\partial \beta_p} \approx \frac{\Delta MSE}{\Delta \beta_p}$$

Where β_p is the shrinkage factor of a single feature p. For sufficiently small $\Delta\beta_p$, the calculated gradient approaches the true gradient. Beta (and sigma) was then updated using the general-purpose gradient descent equation:

$$\beta_f \leftarrow \beta_i - \gamma \nabla \beta$$

where β_i is the initial set of shrinkage factors, $\nabla\beta$ is the calculated partial derivatives, and β_f is the updated set of shrinkage factors. The learning rate γ and initial values for β and σ were tuned empirically.

Gradient descent successfully determines the appropriate level of emphasis on nearest neighbors and important features to minimize cost on the validation set (**Figure 11**). However, this comes at the cost of very large computational time and without much accuracy improvement (if any) when applied to the testing set. Therefore, even though gradient descent approaches proved interesting, we found standard feature selection methods and grid search-based tuning of K as more practical and computationally efficient.

KNN Optimization: Distance metrics

(Onken) We considered different p-norms in the K-nearest neighbors distance calculation. We maintain a C-index of around 0.8 for most norms (**Figure 11**). However, for higher values of p (as $p \to \infty$), we see very poor performance. Since the C-index is based on patient-pair comparisons, one might argue that guessing the opposite of what the C-index guesses results in correct ordering of patients. Nonetheless, since mean-squared errors also had very high values for very large p-norms, we suspected that some numerical or approximation issues were interfering with correct behavior and decided to use the 1-norm for the rest of the experiments.

References

[1] National Institute of Health. The cancer genome atlas, 2016.

[2] American Society of Clinical Oncology. Brain tumor: Statistics, June 2016.

[3] The Cancer Genome Atlas Research Network. Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas. *New England Journal of Medicine*, 372(26):2481–2498, 2015. PMID: 26061751.

[4] Roel G.W. Verhaak et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and {NF1}. *Cancer Cell*, 17(1):98 – 110, 2010.

[5] Michele Ceccarelli et al. Molecular profiling reveals biologically discrete subsets and pathways of progression in diffuse glioma. *Cell*, 164(3):550–563, 2016/12/12.

[6] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

[7] David N. Louis et al. The 2016 world health organization classification of tumors of the central nervous system: a summary. *Acta Neuropathologica*, 131(6):803–820, 2016.

[8] Fatima Cardoso et al. 70-gene signature as an aid to treatment decisions in early-stage breast cancer. *New England Journal of Medicine*, 375(8):717–729, 2016. PMID: 27557300.

[9] Mark A. Hall and Lloyd A. Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, pages 235–239. AAAI Press, 1999.

[10] Yvan Saeys et al. *Robust Feature Selection Using Ensemble Feature Selection Techniques*, pages 313–325. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[11] Yanying Yang. Neural Network Survival Analysis (master's thesis). 2010.

[12] Harald Steck et al. On ranking in survival analysis: Bounds on the concordance index. *Advances in neural information processing systems*, pages 1209–1216, 2008.

[13] D. J. Lowsky et al. A K-nearest neighbors survival probability prediction method. *Stat Med*, 32(12):2062–2069, May 2013.

[14] J. T. Rich et al. A practical guide to understanding Kaplan-Meier curves. *Otolaryngol Head Neck Surg*, 143(3):331–336, Sep 2010.

[15] J. Qian et al. Glmnet for Matlab, 2013.

[16] LeviA. Garraway and EricS. Lander. Lessons from the cancer genome. *Cell*, 153(1):17 – 37, 2013.

[17] Bert Vogelstein, Nickolas Papadopoulos, Victor E. Velculescu, Shibin Zhou, Luis A. Diaz, and Kenneth W. Kinzler. Cancer genome landscapes. *Science*, 339(6127):1546–1558, 2013.

[18] S. H. Park, T. Maeda, G. Mohapatra, F. M. Waldman, R. L. Davis, and B. G. Feuerstein. Heterogeneity, polyploidy, aneusomy, and 9p deletion in human glioblastoma multiforme. *Cancer Genet. Cytogenet.*, 83(2):127–135, Sep 1995.

[19] J. J. Molenaar, M. E. Ebus, J. Koster, E. Santo, D. Geerts, R. Versteeg, and H. N. Caron. Cyclin D1 is a direct transcriptional target of GATA3 in neuroblastoma tumor cells. *Oncogene*, 29(18):2739–2745, May 2010.

Handpicked features									
Model	***Brain Cancer - Basic***					***Breast Cancer - Basic***			
Features	Clinical: Age at initial pathologic diagnosis Mutation: IDH1 IDH2 Chromosomal CNV: 1p 19q					Clinical: Age at initial pathologic diagnosis Histological type is mucinous carcinoma Pathological stage is stage IV No of lymph nodes +ve by H&E Pathologic M is M1 Mutation: ERBB2 mRNA: ALDH4A1; BBC3; WISP1; TGFB3; RAB6B; MMP9; OXCT1; GSTM3; GNAZ; FLT1; EXT1; STK32B; ECT2; GMPS; CDC42BPA			
N x P	779 x 5					815 x 20			
Optimizing deletion thresholds by feature type									
Dataset	**Brain Cancer**							**Breast Cancer**	
Model	*Original*	*Reduced*	*GBM*	*LGG*	*IDHwt*	*IDHmut-Codel*	*IDHmut-nonCodel*	*Original*	*Reduced*
Subpopulations	GBM LGG	GBM LGG	Proneural Neural Classical Mesenchymal	IDHwt IDHmut-Codel IDHmut-nonCodel		??			
N x P	1137 x 17,568	486 x 17,484	62 x 17,482	424 x 17,481	74 x 17,481	149 x 17,481	201 x 17,481	1098 x 17,584	772 x 17,528
Optimizing deletion thresholds by sub-population									
	Brain Cancer								
Data	*Original*		*GBM*	*LGG*	*IDHwt*	*IDHmut-Codel*	*IDHmut-nonCodel*		
Subpopulations	GBM LGG		Proneural Neural Classical Mesenchymal	IDHwt IDHmut-Codel IDHmut-nonCodel		??			
N x P	1137 x 17,568		548 x 96	416 x 17,488	72 x 17,488	144 x 17,488	200 x 17,488		

Table 1: **Datasets used in this report** (*Tageldin and He*)

Before Transformation							
Patient	IDH-1	IDH-2	1p	19q	Age	Survival Time (Added Input)	Censor
A1	1	1	1	0	44	3	1
A2	1	0	0	1	57	4	0

After Transformation							
Patient	IDH-1	IDH-2	1p	19q	Age	Survival Time (Added Input)	Target Status
A1	1	1	1	0	44	1	0
A1	1	1	1	0	44	2	0
A1	1	1	1	0	44	3	0
A2	1	0	0	1	57	1	0
A2	1	0	0	1	57	2	0
A2	1	0	0	1	57	3	0
A2	1	0	0	1	57	4	1

Table 2: **The First data transformation used for the neural network** We replicated features for each year of survival and assigned a label indicating whether the patient was alive or dead at that time. A patient that died was not replicated after the time of failure[11] (*Haque*)

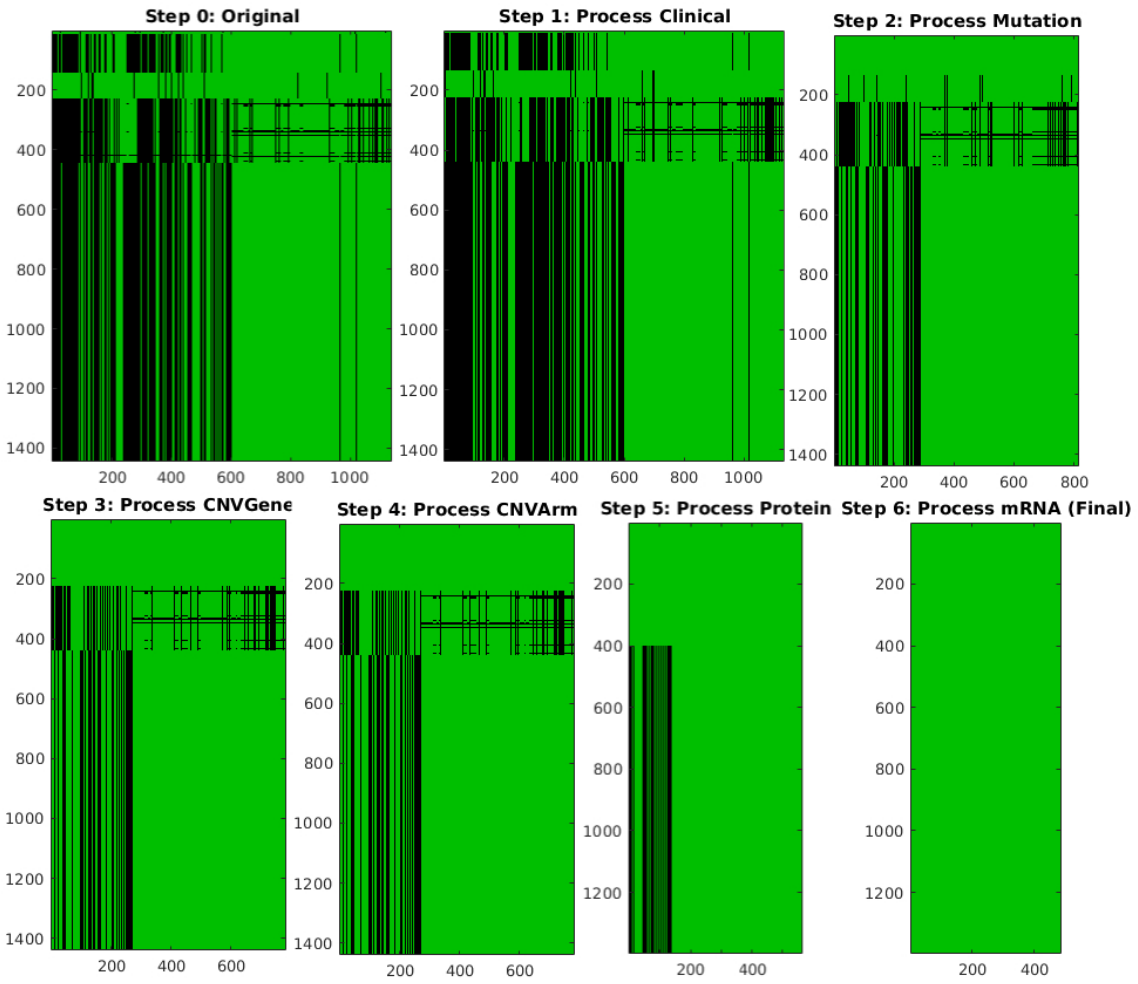

Figure 1: **Data visualization at each step of the preprocessing.** Features are represented in rows and patients in columns. Only the first 1,000 mRNA features are shown for clarity. Available data depicted in green while missing data depicted in black.

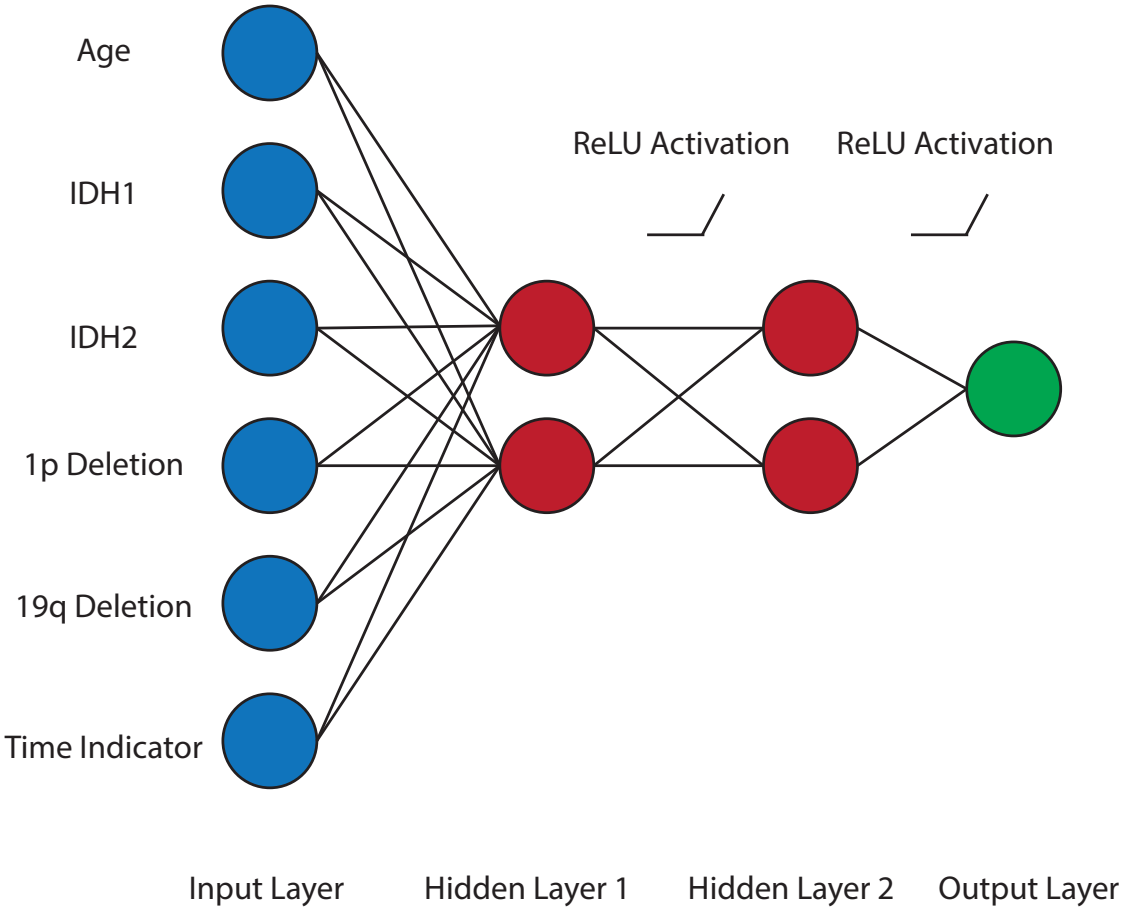

Figure 2: **Artificial Neural Network Architecture.** Our architecture consisted of an input layer with the features of the Basic Model, two hidden layers each with a ReLU activation function, and an output layer.

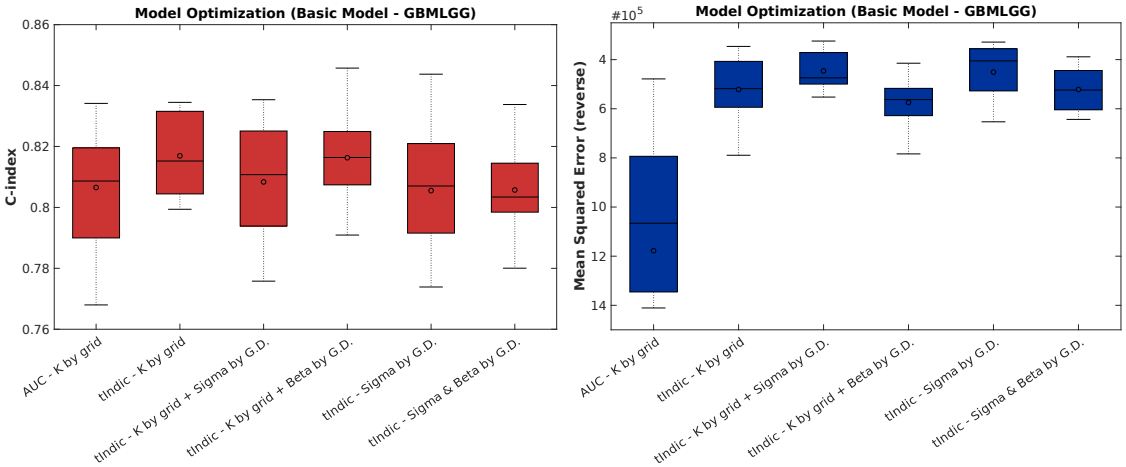

Figure 3: **Optimizing KNN - accuracy of various implementations on GBMLGG Basic model.** Abbreviations used: AUC - area under K-M curve; tindic - time indicator method (non-cumulative probability); K by grid - choosing optimum K using grid search; Sigma - sigma of Gaussian filter used to emphasize closer neighbors; Beta - feature shrinkage vector; GD - gradient descent.

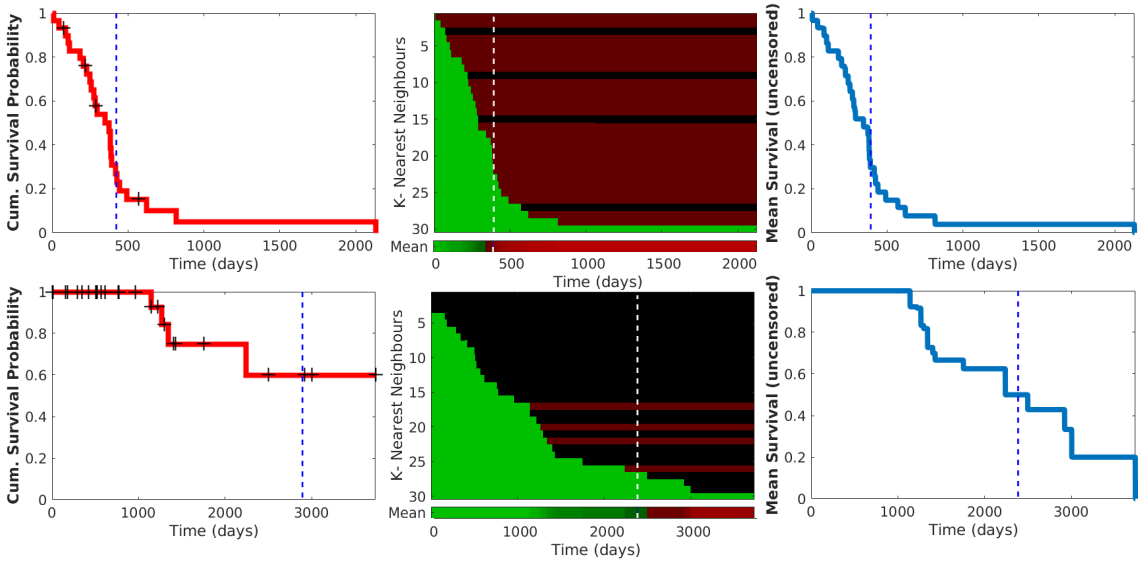

Figure 4: **Optimizing KNN - dealing with censorship. Left:** Kaplan-Meier plots for the K- nearest neighbors of two patients. The vertical dashed line indicates the predicted survival of the "central" patient calculated by integrating the AUC. **Middle** Visual representation of the K-nearest neighbors' survival status at each time from event discovery till the maximum reported survival/censorship time. Each patient is represented in a single row, where green indicates being alive, red indicates being dead, and black indicates unknown status. The survival status of the central patient at each time t is the mean status of uncensored cases at t (bottom "mean" bar). **Right** Predicted survival status of the central patient using the second (non-cumulative probability) implementation. Note the difference in predicted survival from K-M curves when censorship is high (bottom).

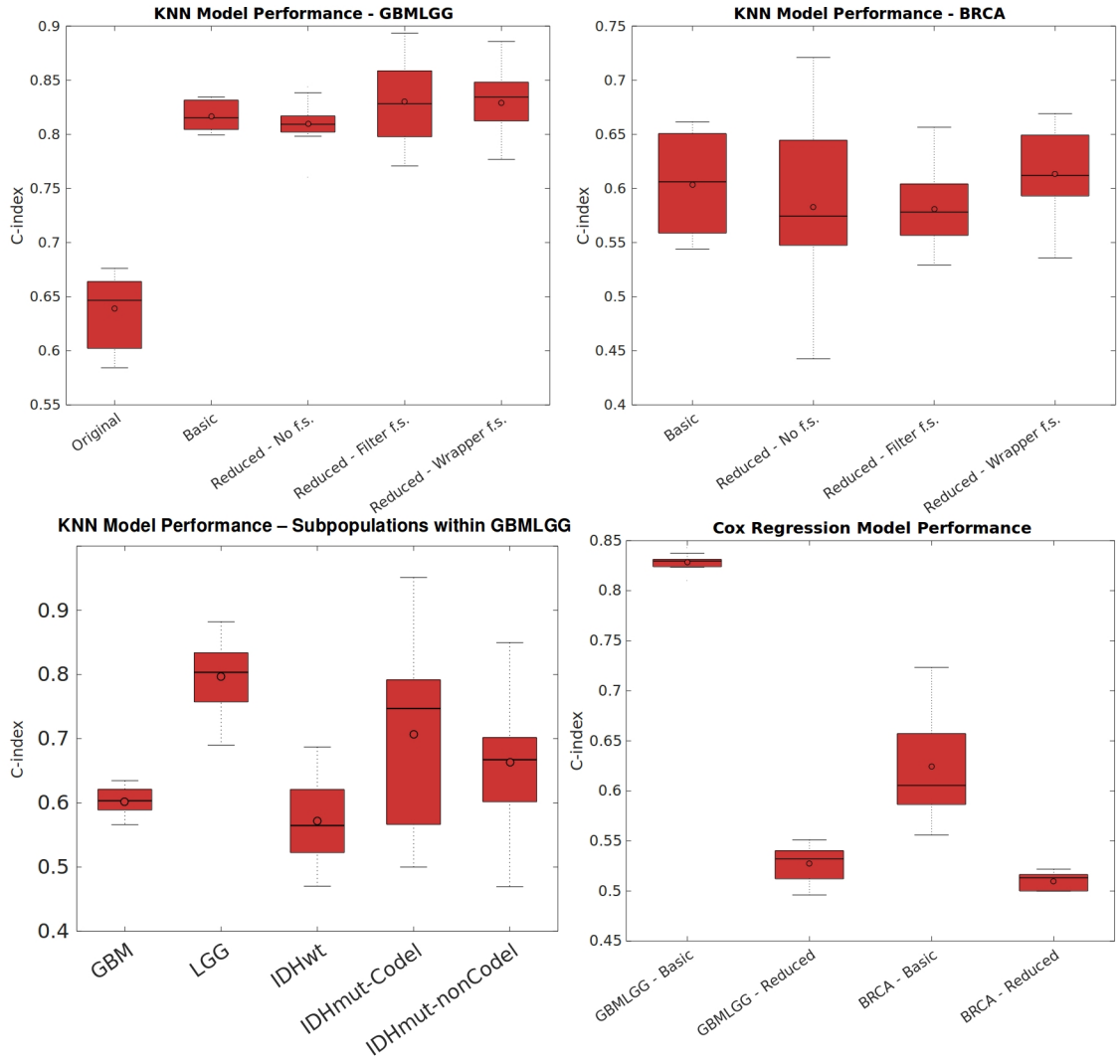

Figure 5: **Model performance of KNN and Cox PH Regression.** Abbreviations used: fs - feature selection. For Cox PH regression, Elastic-Net regularization was applied for the “Reduced” models. Note the marked degradation in accuracy with very high-dimensionality in Cox regression, as opposed to KNN. Wrapper feature selection was used in the KNN implementation for the GBMLGG sub-populations.

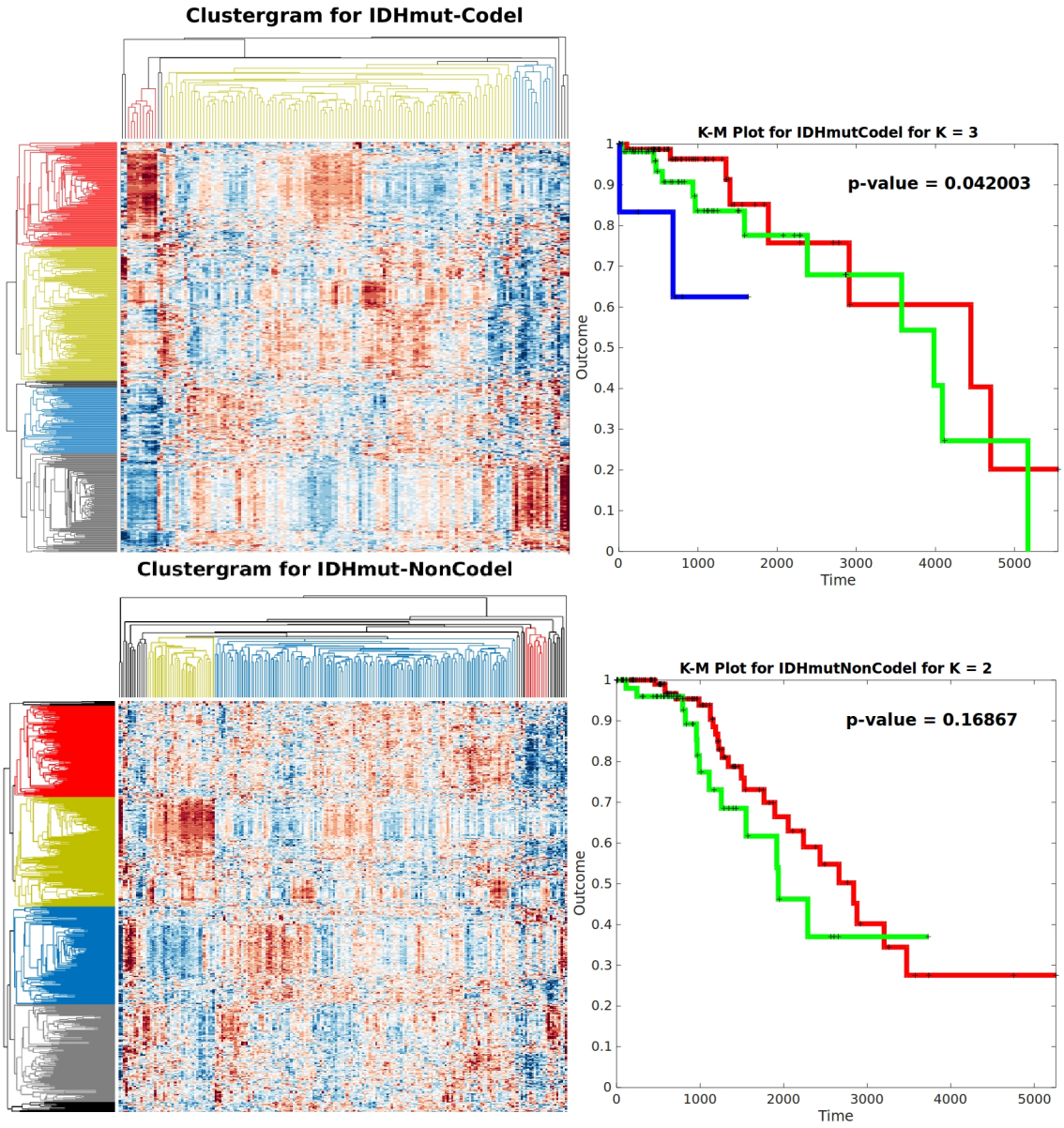

Figure 6: **Cluster-of-Clusters (CoC) Analysis of IDHmut-Codel and IDHmut-NonCodel patients. Left:** The 500 most-varying features were used to generate the clustergrams, where patients are displayed in columns while features are in rows. Red indicates higher feature values, while blue indicates lower feature values. **Right:** K-means clustering was then used to generate potential patient sub-populations (using the same features used in the CoC analysis) to generate K-M plots. P-values were calculated using the Log-Rank test.

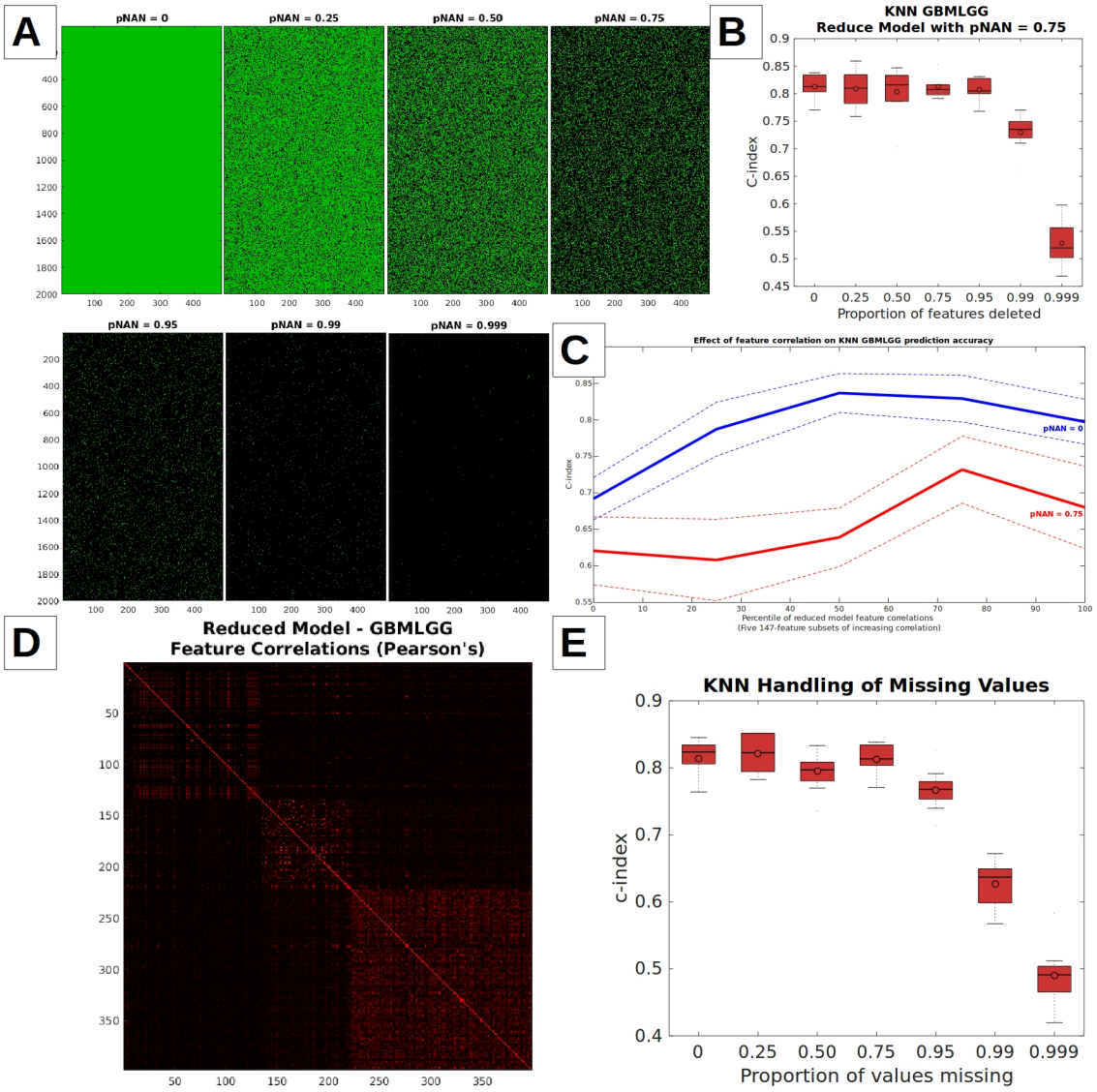

Figure 7: **Moderate feature correlation is a favorable property in handling missing data. A.** Missing data simulation experiment illustration. Features are represented in rows and patients in columns. Abbreviations used: pNAN - proportion of values replaced with NaN (Not-A-Number). **B.** Missing data simulation experiment result. **C.** Effect of feature correlation on KNN prediction accuracy using GBMLGG dataset. Various feature subsets were chosen from based on their pairwise correlations. **D.** Pairwise feature correlations in the GBMLGG dataset. Red indicates higher values. mRNA features were not depicted for clarity. **E.** Testing GBMLGG feature redundancy. Features were randomly deleted from the GBMLGG model that already had 75% of its values randomly deleted.

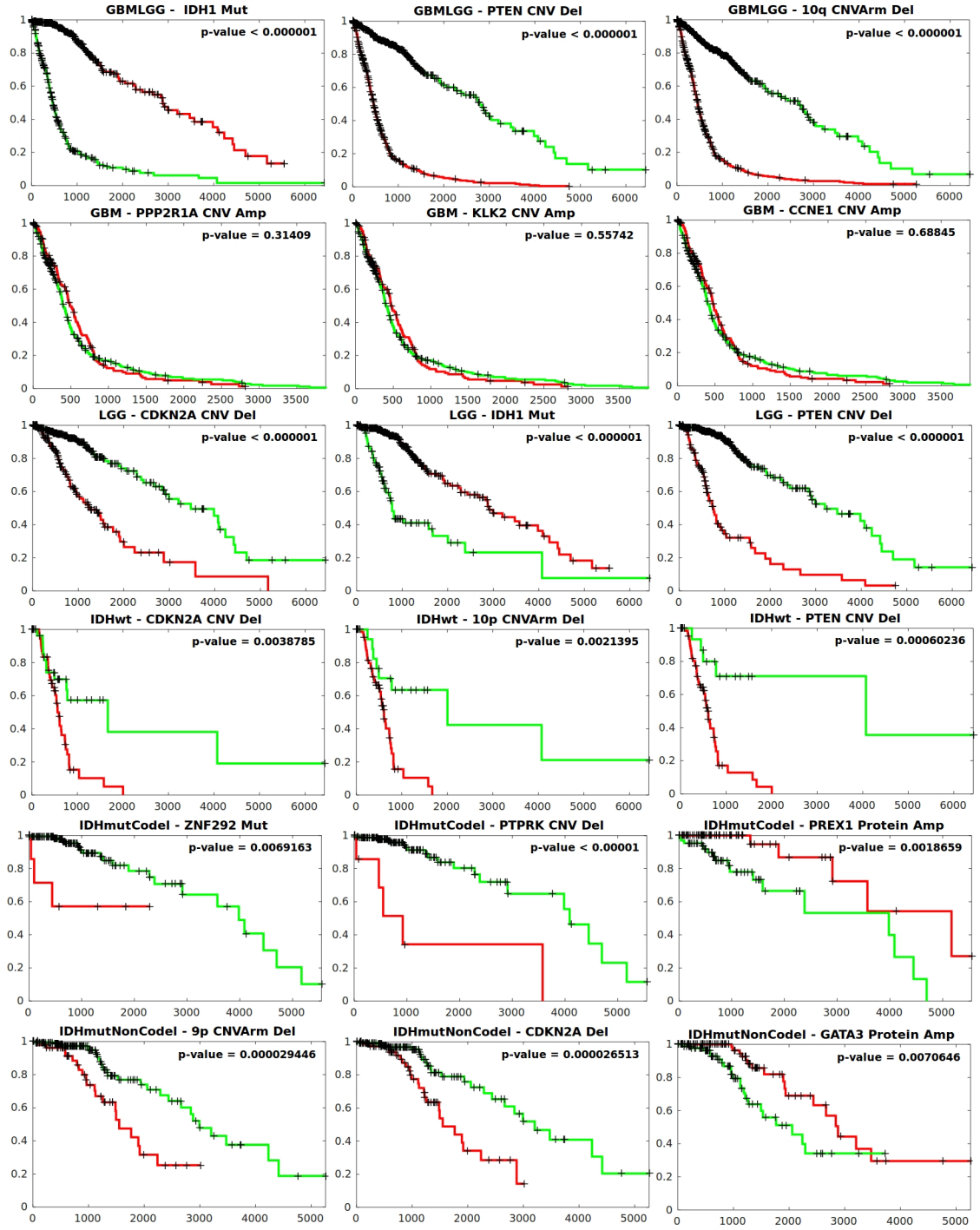

Figure 8: **Kaplan-Meier Plots for the top features selected by the KNN algorithm in various brain cancer sub-populations.** From left-to-right are the first, second and third top features -respectively-, and from top-to-bottom are the brain cancer sub-populations. The x-axis shows the time in days, while the y-axis shows the cumulative survival probability. The red curves depict sub-populations having the feature of interest (eg. gene mutation or amplification), while the green curves depict those who do not. P-values were calculated using the Log-Rank test.

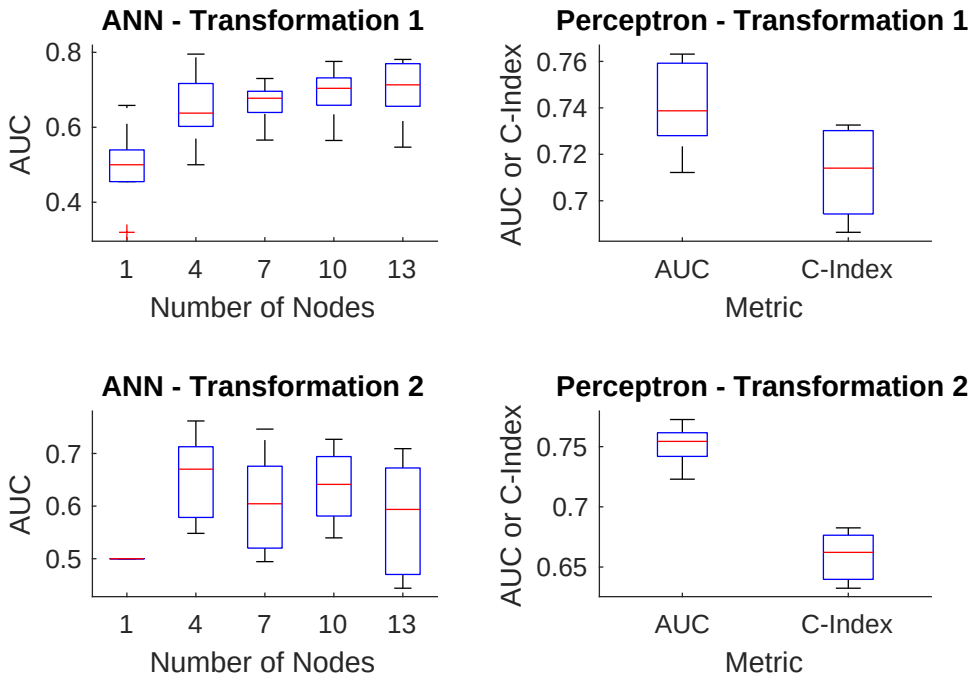

Figure 9: **ANN and Perceptron Performance on GBMLGG Dataset (Basic Model). Left:** The average area under the curve across five folds for the GBMLGG dataset as a function of the number of hidden nodes for each transformation. **Right:** The comparison of perceptron performance for each transformation with the area under curve and C-index metrics.

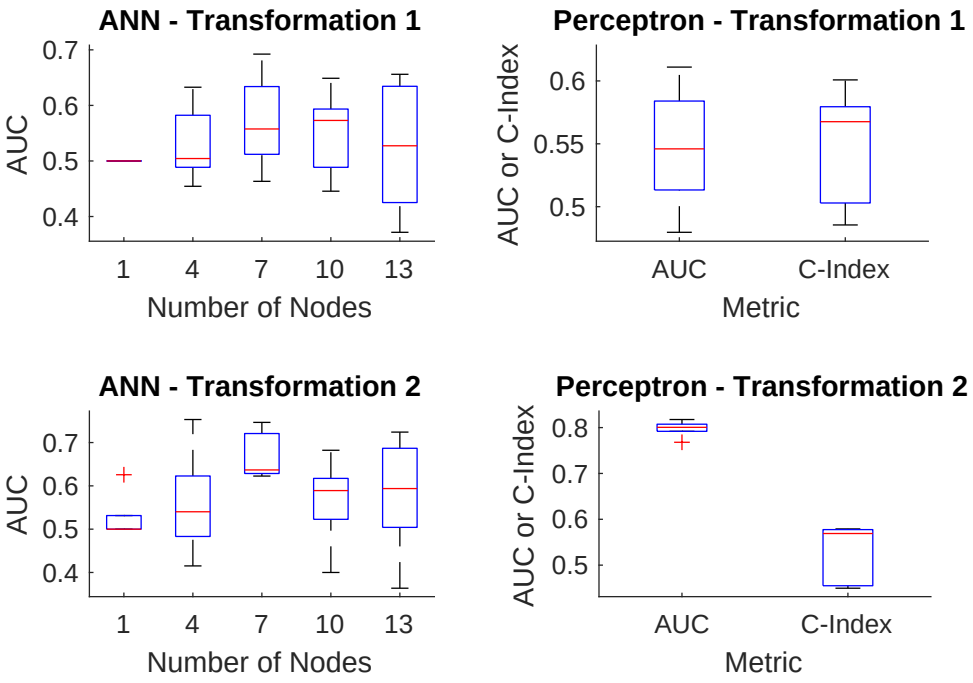

Figure 10: **ANN and Perceptron Performance on BRCA Dataset (Basic Model). Left:** The average area under the curve across five folds for the BRCA dataset as a function of the number of hidden nodes for each transformation. **Right:** The comparison of perceptron performance for each transformation with the area under curve and C-index metrics.

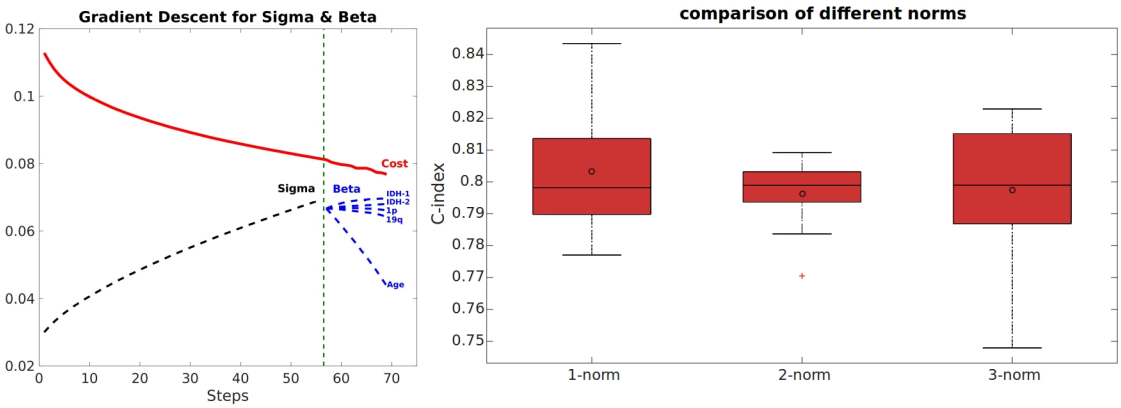

Figure 11: **Optimizing KNN - gradient descent and distance equation. Left:** Emphasizing near neighbors and important features using gradient descent. The cost shown is mean-squared error on the validation set. In this experiment, gradient descent was performed on sigma first, until an optimum value was reached (green vertical line), then gradient descent was performed on beta. Note how age is relatively de-emphasized, indicating its importance. **Right:** Comparison of different p-norms on KNN testing accuracy (GBMLGG).