ABSTRACT

SUBGROUP ANALYSIS BASED ON PROGNOSTIC AND PREDICTIVE GENE SIGNATURES FOR ADJUVANT CHEMOTHERAPY IN EARLY-STAGE NON-SMALL-CELL LUNG CANCER PATIENTS

By

Dustin Pluta

May 2015

In treating patients diagnosed with Stage I non-small-cell lung cancer, doctors must choose between surgery and Adjuvant Cisplatin-Based Chemotherapy (ACT). For patients with resected stages IB to IIIA, clinical trials have shown a survival advantage from 4-15% with the adoption of ACT. However, due to the inherent toxicity of chemotherapy, it is necessary for doctors to identify patients whose chance of success with ACT is sufficient to justify the risks. This project seeks to use gene expression profiling in the development of a statistical decision-making algorithm to identify patients whose survival rates will improve from ACT treatment. Using data from the National Cancer Institute, the Cox-Proportional-Hazards regression model will be used to determine a feasible number of genes that are strongly associated with the treatment-related patient survival. Considering treatment groups separately, patients are assigned a risk category determined by survival time. These risk categories are used to develop a

random forest classification model to identify patients who are likely to benefit from chemotherapy treatment. The probability of significant benefit from chemotherapy is then predicted using a regression survival tree. This model allows the prediction of a new patient's prognosis and the likelihood of survival benefit from ACT treatment based on a small number of gene expression levels.

# SUBGROUP ANALYSIS BASED ON PROGNOSTIC AND PREDICTIVE GENE SIGNATURES FOR ADJUVANT CHEMOTHERAPY IN EARLY-STAGE NON-SMALL-CELL LUNG CANCER PATIENTS

A THESIS

Presented to the Department of Mathematics and Statistics

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Applied Statistics

Committee Members:

Hojin Moon, Ph.D. (Chair)
Sung Eun Kim, Ph.D.
Yong Hee Kim-Park, Ph.D.

College Designee:

Tangan Gao, Ph.D.

By Dustin Pluta

M.A., 2008, University of California, Davis

May 2015




ACKNOWLEDGEMENTS

I would like to thank Dr. Hojin Moon for his support in the completion of my thesis and for his academic guidance. I would also like to thank my committee members Dr. Sung Kim and Dr. Yong Hee Kim-Park.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

# CHAPTER 1

## INTRODUCTION

In a clinical setting, doctors must frequently weigh both conventional and non-conventional medical options to optimize the quality of life and chances of survival for their patients. For patients currently diagnosed with Stage I non-small-cell lung cancer (NSCLC), the standard treatment remains as surgery alone. However, it has been statistically proven that 30 to 40% of this sample population will relapse, demonstrating that patients with a poorer prognosis would benefit from Adjuvant Cisplatin-Based Chemotherapy (ACT) (Zhu et al. 2010). Additionally, current practice is to treat all Stage II patients with ACT, but it is not evident that all such patients receive benefit from ACT.

The adoption of ACT for patients with resected stages IB to IIIA NSCLC was greatly boosted by clinical trials which showed a 5-year survival advantage ranging from 4% in the International Adjuvant Lung trial to 15% in National Cancer Institute of Canada Clinical Trials (Zhu et al. 2010). However, the toxicity of ACT is inevitable. The goal of the present analysis is to use gene expression profiling to identify stage-independent groups of patients who will benefit from ACT, and distinguish other groups of patients who will need to discontinue or forgo ACT treatment to avoid unnecessary toxicity. This will lead to a more accurate treatment decisions, thereby improving the efficacy of ACT treatment.

Numerous previous studies have sought to identify prognostic gene signatures in NSCLC patients (Lu et al. 2006; Raponi et al. 2006; Chen et al. 2007). In particular, the study by Zhu et al. (2010) identified a 15 gene signature able to classify NSCLC patients into high-risk and low-risk groups from four independent data sets. The results of their analysis showed that high-risk patients who received ACT resulted in an improved prognosis, while the low-risk group showed no benefit from ACT, with the treatment possibly having a detrimental effect. The effectiveness of the gene signature found by Zhu et al. strengthens the possibility of developing personalized treatment plans for cancer patients from patient-specific genetic data.

We here apply a modification of the "virtual twins" regression (VTR) method proposed by Foster, Taylor, and Ruberg (2011) for the identification of a subgroup of patients who are at greater likelihood of receiving significant survival benefit from ACT treatment.

As part of the first stage of this method, we develop a random forest classification algorithm of NSCLC patients based on microarray data of gene expression levels that is prognostic of survival for patients receiving surgery only, and also predictive of benefit from ACT treatment in the JBR.10 data set, which is the result of a randomized phase III clinical trial for comparing ACT treatment against surgery alone. After detailing the construction of the development of the patient categorizations and random forest model, we summarize the classification as an algorithm for determining whether a patient should receive ACT treatment. Model accuracy is estimated via cross validation and accuracy for prognosis of survival without ACT treatment is further validated through application to an independent test set that was generated to validate the results of Zhu et al (2010).

The second stage of the VTR method uses treatment effect estimates calculated in the first stage as response variables in a regression tree, with gene expression levels as predictors. From the regression tree, a region of enhanced treatment effect is identified as a subset of the predictor space, defined by a small number of genes. The magnitude of the enhanced treatment effect for this region is estimated using the bias-corrected bootstrap method proposed by Foster, Taylor, and Ruberg (2011).

# CHAPTER 2

# STATISTICAL BACKGROUND

## 2.1 CART Algorithm

The classification and regression tree (CART) algorithm introduced by Breiman et al. (1984) offers a nonparametric approach to classification and regression problems, and can also be adapted for survival data. Considering a classification problem with $k$ classes, $p$ predictors, and $n$ observations, the CART algorithm begins with a root node containing all observations. This node is split into two child nodes according to a splitting criterion determined based on one predictor such that the split minimizes an impurity measure. For classification problems, the impurity measure is the Gini impurity index, which is calculated as $i(t) = 1 - \sum_k p^2(k|t)$ for a node $t$ (Ahn and Moon 2010). Nodes are recursively split by choosing the next split to maximize the reduction in impurity, defined as $\Delta i(\theta, t) = i(t) - [p_l i(l) + p_r i(r)]$, where $\theta$ is a split of node $t$ into nodes $l$ and $r$ with a proportion $p_l$ of observations in $t$ going to $l$, and proportion $p_r$ going to $r$ (Ahn and Moon 2010). Splitting continues until a stopping criterion is reached. The default stopping criterion suggested by Breiman et al. (1984) is terminate the algorithm when all nodes either contain a single class or have size $N(t) \leq 5$ for all nodes $t$, where $N(t)$ is the number of observations in $t$; the R implementation of CART in the package *rpart* uses a default stopping criterion of $N(t) = 20$. The optimal

minimum node size can be determined using cross-validation. For prediction in classification problems, each terminal node (also known as leaf nodes) in the resulting tree is assigned an overall class equal to the majority class in the node. The predicted class of a new observation is the class of the terminal node that the new observation belongs to. For regression and survival data, the CART algorithm can be applied using different impurity measures, typically mean-square error for regression, and the log-rank test statistic for survival data.

One potential problem of the CART algorithm is overfitting the training data, which occurs when trees are grown too deeply, resulting in decision boundaries that closely track the classes of the training sample. A common approach to avoiding this problem is the application of cost-complexity pruning. For a tree $T$, let $R(T)$ be an estimate of the tree misclassification rate, and $\left|\tilde{T}\right|$ be the number of terminal nodes in the tree. For a predefined complexity parameter $\alpha$, the cost-complexity measure $R_\alpha(T)$ is defined as

$$R_\alpha(T) = R(T) + \alpha\left|\tilde{T}\right|.$$

Starting with a deeply grown tree, cost-complexity pruning constructs a nested sequence of subtrees by successively deleting branches to minimize $R_\alpha(T)$ (Tibshirani, Friedman, and Hastie, 2009). The resulting tree with minimum $R_\alpha(T)$ will producer coarser decision boundaries (due to fewer terminal nodes) and thus be less prone to overfitting.

A related problem for CART is high-variance in the tree structure with respect to changes in the sample data (Bou-Hamad, Larocque, and Ben-Ameur 2011). The decision boundaries produced by the CART algorithm are rectilinear boundaries parallel to the

variable axes. Small variations in the data can produce changes in the tree structure by causing splits with different predictors or introducing additional splits. Changes in the tree structure are likely to result in large shifts in the decision boundaries, with the result that small changes in the training data can result in significantly different predictions. This high-variance is of particular concern when working with noisy data or data with many predictors that are not strongly correlated with the response, as is the case with microarray data.

## 2.2 Random Forests

Ensemble methods such as bootstrap-aggregating ("bagging"), random forests, and boosting have been shown to effectively reduce the variance of CART and improve the overall predictive accuracy (Tibshirani, Friedman, and Hastie 2009; Bou-Hamad, Larocque, and Ben-Ameur 2011). Boosting is the basis for the effective AdaBoost algorithm (Tibshirani, Friedman, and Hastie 2009), which grows a sequence of small trees, with each tree fitted to the residuals of the previous tree, instead of the response itself. Alternatively, bagging produces an ensemble of trees, each grown from a bootstrapped sample of the original training data. From this ensemble, the predicted class of a new observation is determined by majority vote of the predictions of the trees in the ensemble. Random forests are a modification of the bagging method that improves performance through decorrelating the trees in the ensemble (Breiman 2001).

The bagging method grows each tree by considering all $p$ predictors from the data set, consequently the structures of the trees in the ensemble are highly correlated. Because of this correlation, the most significant predictors are likely to occur in many

trees in the ensemble, and thus bagging may still be susceptible to high-variance and dependency on the training data. For the random forest method proposed by Breiman (2001), instead of considering all predictors for every tree, each tree is grown considering a random selection of $m$ predictors from the total set of predictors. By randomly selecting a subset of predictors, the correlation of the trees in the ensemble is reduced, leading to a greater reduction in variance for the random forest model compared to simple bagging.

To see how decorrelation improves performance, consider a sample of $B$ identically distributed (but *not* independent) random variables $X_i, i = 1, \dots, B$, with variance $\sigma^2$ and positive pairwise correlation $\rho$. The variance of the average is

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = \frac{1}{B^2}\left[\sum_{i=1}^{B} Var(X_i) + 2\sum_{i=1}^{B}\sum_{j=1}^{i-1} Cov(X_i, X_j)\right]$$

$$= \frac{1}{B^2}\left[B\sigma^2 + 2\rho\sigma^2\sum_{i=2}^{B}(i-1)\right]$$

$$= \frac{\sigma^2}{B^2}\left[B + 2\rho\left(\frac{B(B-1)}{2}\right)\right]$$

$$= \sigma^2\left[\frac{1}{B} + \rho\frac{(B-1)}{B}\right]$$

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Since the variance of the average performance of an ensemble of predictions increases as the correlation of the predictions increases, the benefits of averaging the bagged trees is limited by the correlation of the trees.

As an additional advantage over CART, bagging, and boosting, Breiman (2001) proved random forests do not overfit the data, even for a very large number of trees. Random forest decision boundaries tend to be axis-oriented due to the nature of the tree

decision boundaries, but the ensemble voting allows for much more dynamic boundaries than sharp rectilinear edges. A thorough analysis of the operation of random forests is given in Tibshirani, Friedman, and Hastie (2009).

The main parameters for the random forest method are the number of trees to grow, $ntree$, and the number of predictors to try per tree, $m$. These values can be chosen to minimize the estimated classification error using cross-validation. The random forest method is implemented in the R package *randomForest*; for classification problems the default parameters are $ntree = 500, m = \sqrt{p}$. These values have been shown to have consistently good results across a variety of problems (Ahn and Moon 2010).

## 2.3 Virtual Twins Regression

Virtual twins regression is a statistical approach to the problem of estimating the effect of a treatment on a population of patients proposed by Foster, Taylor, and Ruberg (2011) (e.g., lung cancer patients undergoing chemotherapy). Assume our data consists of $n = n_0 + n_1$ total patients, with $n_0$ control group patients and $n_1$ treatment group patients, such that each patient $i$ is assigned a treatment $T_i \in \{0, 1\}$, with $T_i = 0$ corresponding to the control group, $T_i = 1$ corresponding to the treatment group. Let $Y_i \in \{0, 1\}$ be the response for patient $i$; we here consider $Y_i = 1$ to be a negative disease outcome, such as death. A region of enhanced treatment effect $A$ is a region of the predictor space such that patients who are classified in $A$ are at greater likelihood of benefit from the treatment $T_i = 1$.

Virtual twins regression estimates this region by $\hat{A}$ with a two-stage model. The first stage of VTR builds independent random forests from both treatment groups. The

treatment effect estimate $Z_i$ for patient $i$, with predictors $X_i$ is $Z_i = \hat{P}_{1i} - \hat{P}_{0i}$, where $\hat{P}_{0i} = P(Y_i = 1 \mid T_i = 0, X_i)$ and $\hat{P}_{1i} = P(Y_i = 1 \mid T_i = 1, X_i)$. If the true treatment for patient i is $T_i = j$, then $\hat{P}_{ji}$ is estimated by the out-of-bag estimate from the random forest for $T = j$, and $\hat{P}_{(1-j)i}$ is estimated by applying the random forest for $T = 1 - j$ and setting $\hat{P}_{(1-j)i}$ equal to the percentage of votes that predict $Y_i = 1$.

These estimated treatment effects are then used as response variables in a regression tree, trained on the union of treatment groups, and including all predictors considered in the random forests. The terminal nodes with predicted effect greater than some threshold c define the estimated enhanced treatment effect region $\hat{A}$, which will depend on a small number of predictors. Foster, Taylor, and Ruberg (2011) recommend picking $c = \delta - 0.05$ or $c = \delta - 0.1$, where $\delta = P(Y = 1 \mid T = 1) - P(Y = 1 \mid T = 0)$ is the estimate of the population treatment effect. In the sequel we adopt $c = \delta - 0.05$.

## 2.4 Bias-corrected Bootstrap

The enhanced treatment effect of the VTR model can be estimated using the bias-corrected bootstrap method, which has been shown to produce more accurate estimates in most instances compared to competing methods (Foster, Taylor, and Ruberg 2011). We first examine the general structure of the bias-corrected bootstrap method, followed by the specific application to the VTR model. Consider $n$ discrete random variables $X = (X_1, X_2, \dots, X_n)$, with unknown true distribution $p(X)$. For given data $D$ sampled from $X$, the basic bootstrap procedure constructs $B$ bootstrap samples $D^{(b)}, b = 1, \dots, B$. Define the empirical distribution of $D$ as $p(x) = n(x)/n$, where $n(x)$ is the frequency of

$x$ in $D$. From these bootstrapped samples, a statistic $T(X)$ can be estimated by $\hat{T}(X) = \frac{1}{B}\sum_{b=1}^{B} T(D^{(b)})$, where $T(D^{(b)})$ is computed from the empirical distribution $\widehat{p_b}$ of $D^{(b)}$. The estimated bias of the statistic $T$ is $\widehat{\text{Bias}_T} = E_b\left[T(D^{(b)})\right] - T(\hat{p})$. The bias-corrected bootstrap estimate of the statistic $T$ is then calculated from the sample $D$ as $T^{BC}(D) = T(D) - \text{Bias}_T$ (Steck and Jaakkola 2003; Efron and Tibshirani 1993).

Foster, Taylor, and Ruberg (2011) adapt the bias-correct bootstrap method given above to the case of estimating the treatment effect of the estimated enhanced treatment effect region. For a region of the predictor space $R$, let the enhanced treatment effect measure for region $R$ be defined as

$$Q(R) = \big(P(Y = 1|T = 1, X \in R) - P(Y = 1|T = 0, X \in R)\big) -$$

$$(P(Y = 1|T = 1) - P(Y = 1|T = 0)).$$

For the estimated enhanced treatment effect region $\widehat{\text{A}}$, $\text{Q}\big(\widehat{\text{A}}\big)$ is the estimate of the treatment effect for this region. The bias corrected bootstrap procedure first creates 20 bootstrap resamplings (with replacement) of the original data, with each sample containing $n_0$ patients sampled from the control gorup and $n_1$ patients sampled from the treatment group. The estimate $\text{Q}\big(\widehat{\text{A}}\big)$ is computed for each bootstrap sample by (Q from original data applied to original $\widehat{\text{A}}$) + (Q from original data applied to new $\widehat{\text{A}}$) – (Q from new data applied to new $\widehat{\text{A}}$), and the results of these sample computations are averaged for the final estimate (Foster, Taylor, and Ruberg 2011).

# CHAPTER 3

# CLINICAL BACKGROUND

Due to the ineffectiveness of individual genes as prognostic markers (Zhu, et al. 2010), researchers have turned to multi-gene signatures as a means of improving predictive power in identifying those patients with a poor prognosis, for whom ACT could be considered. Previous studies have applied the classification and regression tree (CART) algorithm developed by Breiman et al. (1984) to cancer patient data. CART has the advantage of producing an easily interpretable model from which one can identify a multi-gene signature.

Hess et al. (1999) applied the CART algorithm to predict survival time of 1000 patients with unknown primary carcinoma based on 26 clinical factors. The CART algorithm was chosen for its ability to identify prognostic subgroups by classifying patients based explicitly on the covariates, in a manner that can easily be adapted for practical clinical use. The resulting classification tree determined by the basic CART algorithm produced 10 terminal subgroups of patients with similar estimated survival characteristics. To examine the effect of alternative splits on the structure of the tree, two additional trees were constructed from 500 bootstrapped samples generated from the original data. The results of the bootstrap analysis showed that the CART algorithm consistently chooses important covariates for the initial split; a primary split on liver involvement was chosen for 41% of samples, a primary split on histology type was

chosen for 27% of samples, and a primary split on lymph node involvement was chosen for 23% of samples. Hess et al (1999) remarked that visual inspection of the three trees indicated that the overall tree structure might be driven by interactions among the covariates. This suggests that a potential drawback of the CART procedure is an inability to model all significant interactions with appropriate weight.

Despite the flexibility of the CART algorithm, studies (Bou-Hamad et al. 2011; van Dijk et al. 2004) have shown it to exhibit a sensitivity to the characteristics of the training sample, particularly for small sample sizes. Kollmannsberger et al. (2000) apply CART to identify poor prognosis subgroups among 332 nonseminomatous germ cell cancer patients. To evaluate the performance of the Kollmansberger tree, van Dijk et al. (2004) applied the classification scheme to 456 poor prognosis patients in the International Germ Cell Consensus Collaborative Group (IGCCCG) patient database. Clinical covariates and follow-up times for the Kollmansberger and IGCCCG patients were similarly distributed; 2-year survival was 56% for the Kollmansberger patients and 72% for the IGCCCG patients. As measured by the concordance statistic (c-statistic)[1], applying the Kollmansberger tree to the IGCCCG patients produced a lower discriminative ability compared to the original data, with c=0.56 on the IGCCCG patients compared to c=0.63 on the original data set. To compare the performance of the Kollmansberger tree, van Dijk et al. (2004) produced a new classification tree from the 456 IGCCCG patients, which exhibited a similar discriminative ability at c=0.59. For

[1] For binary response data, the c-statistic is equivalent to the area under the ROC curve. The c-statistic ranges from 0.5 (no predictive value) to 1.0 (perfect discrimination of differing survival) (Grunkemeier and Jin 2001).

comparison, the discriminative ability for a Cox regression model applied to the IGCCCG data was c=0.61. The two trees showed very different structures and included different covariates for the primary splits, indicating different covariate effects and interactions across the two samples. The significant variation in the trees generated by the two patient sets led van Dijk et al. (2004) to conclude that the generated trees were unsatisfactory for clinical use, and that a more stable method such as Cox regression may be preferred, especially with small data sets.

In attempting to classify patients according to survival characteristics, an additional problem with the application of survival trees is the potential for distinct terminal nodes of the tree to exhibit similar survival profiles. Tsai et al. (2007) proposed a modification to the CART algorithm through agglomerative hierarchical clustering in an attempt to group terminal subgroups with statistically equivalent survival. After building the survival tree, the agglomerative hierarchical clustering procedure successively combines terminal subgroups with the most similar survival profiles as determined by the log-rank test (Tsai et al. 2007). The procedure continues until all statistically similar groups have been combined (according to a pre-specified cutoff for the log-rank *p*-value or simply until all nodes have been merged); the order of grouping the terminal nodes results in a dendrogram (nested clusters) of the initial CART terminal nodes, from which prognostic categories can be formed (e.g., the final two separate groups can be considered as high-risk and low-risk subgroups based on median survival). To evaluate the performance of the integrated tree classification scheme, Tsai et al. (2007) proposed a *k*-fold cross validation scheme wherein the classification of the test

data was compared to the training data via the log-rank test to estimate the efficacy of the classification on independent data. As an illustration of the procedure, Tsai et al. (2007) produced a classification of 13,268 melanoma patients from the American Joint Committee on Cancer (AJCC) Melanoma Database based on six clinical and pathological variables. The initial survival tree produced 11 terminal subgroups; visual inspection of the Kaplan-Meier survival curves of these subgroups revealed some similar survival characteristics across subgroups, illustrating the limitation of the survival tree classification. Applying agglomerative hierarchical clustering with a cutoff of $P$=0.05 merged four subgroups, resulting in seven final subgroups, each with significantly different survival profiles as measured by the log-rank test. The resulting seven subgroups had highly significant differences in survival curves (log-rank $p$-value < 0.0001). The effectiveness of the model was verified through cross-validation analysis, which showed the survival profiles of the test data subgroups were not significantly different from the same subgroups of the training data (Tsai et al. 2007).

The study by Tsai et al. (2007) supported the use of survival trees for subgroup analysis. However, the training set included 13,268 patients, which is a much larger sample size than is typically available. As found by van Dijk et al. (2004), the efficacy of the survival tree method is questionable for smaller data sets. Furthermore, a comparative study by Bou-Hamad et al. (2011) showed that a single survival tree has poor predictive performance relative to more robust methods. In particular, the results of Bou-Hamad et al. showed the random forest algorithm shows the best performance in predicting survival of patients with primary biliary cirrhosis of the liver as measured by

the Integrated Brier Score. By building many bootstrapped survival trees and producing an ensemble prediction, the random forest algorithm is able to reduce the sensitivity to the sample data inherent in a single survival tree model.

A current problem of clinical interest is to identify personalized treatment plans for patients, which requires the identification of patient subgroups with high likelihood of benefit from a given treatment. In considering this problem, Foster, Taylor, and Ruberg (2011) suggested a two-stage model, which first constructs a random forest to estimate the treatment effect for each patient; and second, applies the CART algorithm to the predictors and treatment effect estimates from the first stage to determine a small number of predictors that are associated with the treatment effect. This model was shown to perform well on clinical trial data and simulated data, as verified through many different validation methods (Foster, Taylor, and Ruberg 2011). The simulation study suggested that the most effective validation method of those considered was the bias corrected bootstrap method applied to simulated data generated from the original sample (Foster, Taylor, and Ruberg 2011).

# CHAPTER 4

## PATIENTS AND METHODS

The JBR.10 data set used by Zhu et al. (2010) is the result of a randomized phase III controlled trial of adjuvant vinorelbine/cisplatin chemotherapy versus observation alone (Winton et al. 2005). The goal of the JBR.10 study was to determine whether patients with completely resected non-small-cell lung cancer receive survival benefit from adjuvant vinorelbine plus cisplatin treatment. A total of 482 were randomly assigned to either observation with no chemotherapy, or a regimen of ACT treatment. The median age of patients was 61 years, 65% of patients were men, 53% had adenocarcinomas; approximately 45% of patients were stage IB, 15% were stage IIA patients, and 40% were stage IIB patients. Numerous toxic effects occurred as a result of the chemotherapy treatment in a large portion of the patients under study, including fatigue (81% of patients), nausea (80%), and anorexia (55%). However, sever toxic effects occurred in less than 10% of the ACT patients. The chemotherapy group exhibited significantly prolonged overall survival with median survival at 94 months and 73 months for the chemotherapy and observation groups respectively; the Cox regression hazard ratio of death for chemotherapy vs observation was 0.69, P=0.04 (Winton et al. 2005). Subgroup analysis performed by Winton et al. showed that the greatest survival advantage was exhibited by stage II patients, although they note that the number of stage

IB patients was small, and the stage-by-treatment interaction was not significant (P=0.13). A follow up analysis was conducted by Butts et al. (2009), which showed a continued survival advantage in stage II patients, but no significant survival difference in stage IB patients; a Cox regression model on stage, treatment, and stage-treatment interaction showed borderline significance (P=0.09) (Butts et al. 2009).

Frozen tumor tissue samples were collected from 169 of the 482 patients in the JBR.10 data set. Gene expression profiling was conducted for 133 of these samples by the Affymetrix HG-U133A technology at Center for Cancer Genome Discovery, Dana-Farber Cancer Institute. Of the 133 patients, 71 received adjuvant chemotherapy (ACT), while 62 patients were under observation only (OBS).

## 4.1 Probe Set Preprocessing and Screening

The raw CEL data for this study was obtained from the Gene Expression Omnibus (National Center for Biotechnology Information 2009) and preprocessed using the Robust Microchip Analysis (RMA) method using the R package *Affy* (Bioconductor 2014). The RMA algorithm processes probe set expression values by performing a background adjustment calculated from probe intensities, followed by quantile normalization, and summarization of probe set intensities into a single expression value (Dziuda 2010). The data were then centered to the mean and scaled to unit variance following RMA normalization.

Initial screening of the probe sets is employed using a univariate Cox proportional hazards model and leave-one-out cross-validation significance scores, screening separately for the OBS patients and ACT patients. For each treatment group, one patient

at a time was excluded, and each probe set tested for significance at the $\alpha = 0.05$ level. The significance scores of a probe set were calculated as the number of times the probe set was found to be significant in the cross-validation procedure for each treatment group. Using a minimum score cutoff of 2 for both treatment groups, there were 496 probe sets from the OBS patients, and 406 probe sets selected from the ACT patients. For the selected probe sets, differential expression analysis was performed using the GEO2R tool from the Gene Expression Omnibus to ensure similar expression levels across treatment groups for the selected probe sets (National Center for Biotechnology Information 2014). The GEO2R tool is built using the *Linear Models for Microarray Data* (limma) package (Bioconductor 2013). The limma algorithm calculates an adjusted *t* statistic from a linear model fit to each gene. A two-sample *t* test is then applied with an adjustment for multiple comparisons; we here used the default GEO2R multiple comparison adjustment method, the Benjamini and Hochberg procedure (Dziuda 2010). Visual examination of boxplots of probe set expression values showed average probe set expression values were similarly distributed across all patients (FIGURE 7).

## 4.2 Model Design

### 4.2.1 Summary

We now describe the design of a model to predict a patient's survival benefit from ACT treatment from probe set expression levels, and which identifies a subgroup of patients who are likely to have significant benefit from ACT treatment. Adapting the design suggested by Foster, Taylor, and Ruberg (2011), we build a predictive model composed of two main stages. The first stage consists of a random forest trained on each

treatment group separately. The response variable is the patient treatment recommendation, determined by their time to follow-up or censoring relative to the median of the respective treatment group. We use these random forest predictions to generate a treatment recommendation for a patient based on their microarray data, namely ACT treatment recommended (*CHEMO-REC*) or ACT treatment not recommended (*CHEMO-NOTREC*). This gives a coarse prediction of whether a patient is likely to receive survival benefit from ACT treatment.

The random forests do not provide an estimate of the strength of the treatment effect, nor is it possible to clearly identify subgroups of patients with similar responses to ACT treatment. To refine the random forest model, we follow the "virtual twins" method given by Foster, Taylor, and Ruberg (2011). This method estimates the treatment effect $Z_i$ for patient *i* from the random forests, which can then be used as the response in a regression tree. This tree provides a prediction estimate of a new patient's benefit from ACT treatment, and also identifies a subgroup of patients with a higher likelihood of benefit from treatment.

4.2.2 Prognostic Classification of Patients

In order to train a random forest on these selected probe sets, we applied classification labels to the patients based on their treatment group and survival results. The OBS patients had median overall survival of 3.815 years; the ACT patients had median overall survival of 5.81 years. From this, OBS patients with survival less than 3.815 years were classified as *CHEMO-REC* and OBS patients with survival at least 3.815 years were classified as *CHEMO-NOTREC*. Similarly, ACT patients with survival

less than 5.81 years were classified as *CHEMO-NOTREC*, and ACT patients with survival at least 5.81 years were classified as *CHEMO-REC*. We note that the median survival for ACT patients is 2 years more than that of OBS patients.

TABLE 1. Given Classes for Patients by Treatment Group

| Treatment | Survival | Given Class |
|---|---|---|
| ACT | *OS* < 5.81 yrs | *CHEMO-NOTREC* |
| | *OS* >= 5.81 yrs | *CHEMO-REC* |
| OBS | *OS* < 3.815 | *CHEMO-REC* |
| | *OS* >= 3.815 | *CHEMO-NOTREC* |

4.2.3 Random Forests for Each Treatment Group

Following a modification of the approach used by Foster, Taylor, and Ruberg (2011), we use these classifications and patient microarray data to create two random forests, one for each patient treatment subgroup. The OBS random forest is trained on the OBS patients and significant OBS genes, and classifies patients as *CHEMO-NOTREC* or *CHEMO-REC*, and thus estimates the risk of early death for future patients based if they undergo surgery without chemotherapy treatment. The ACT random forest is trained on the ACT patients and significant ACT genes, and classifies patients as *CHEMO-NOTREC* or *CHEMO-REC*, classifying future patients on their predicted

survival under ACT treatment. From this we derive the predictive categories: when both the OBS and ACT random forests agree in their predictions a patient is classified accordingly as *CHEMO-NOTREC* or *CHEMO-REC*; when the two random forest produce conflicting classifications, the patient is classified as inconclusive, patient's or doctor's choice (*CHOICE*).

The parameters random forests are tuned using leave-one-out cross-validation to estimate the optimal number of trees grown and number of predictors consider per tree. Specifically, the parameter values for number of trees and number of predictors per tree are iterated over a range of values, and model accuracy is evaluated using LOOCV for each of the considered parameter values. The results of this cross-validation suggest using 500 trees and 25 predictors per tree as model parameters.

4.2.4 Enhanced Treatment Effect Subgroup with VTR

To identify a subgroup of patients for whom ACT treatment is likely to be particularly effective, we follow the VTR model by first estimating the treatment effect $Z_i$ for patient i as $Z_i = \widehat{P}_{1i} - \widehat{P}_{0i}$, where $\widehat{P}_{1i}$ is the probability of early death for patient i under ACT treatment, and $\widehat{P}_{0i}$ is the probability of early death under surgery alone. The probabilities $\widehat{P}_{1i}$ and $\widehat{P}_{0i}$ are calculated by applying both random forests to each patient and assigning patient i probabilities according to the percentage of trees that classify a patient as either *REC* or *NOTREC* respectively (using the out-of-bag estimate when applying the random forest of a patients true treatment group).

The treatment effect estimates $Z_i$ are then used as response variables for a regression tree, trained on all patients and the union of predictors from the OBS and ACT

random forests. The regression tree is computed using the R package *rpart*, with minimal terminal node size 20 and complexity parameter 0.02, as suggested by Foster, Taylor, and Ruberg (2011). The terminal nodes (and their associated branches) with estimated treatment effect $< c = \delta - 0.05$ define the estimated enhanced treatment effect region $\widehat{A}$. This region depends on a small number of probe set values, and provides a clearer indication of important genes and their relationships than is possible with the random forests. The bias-corrected bootstrap is then applied to estimate the treatment effect for this region $Q(\widehat{A})$. Patients whose probe set values place them in the enhanced treatment effect region are at a higher likelihood of survival benefit from chemotherapy treatment, with the estimated benefit $Q(\widehat{A})$ equal to the predicted reduction in likelihood of early death under chemotherapy treatment compared to surgery alone.

# CHAPTER 5

# RESULTS

## 5.1 Internal Validation of Random Forest Stage

Due to the random nature of the random forest algorithm (both through bootstrapping samples and from random selection of predictors), every application of the model yields slightly different results. This makes the reporting of performance measures (e.g., accuracy, specificity, sensitivity) slightly more complicated, but this complication is mitigated by the low-variance expected with the random forest model. We present here the results of a "typical" application of the model, followed by the average results of 100 runs of the model.

### 5.1.1 Comparison of Survival Characteristics by Treatment and Model Predictions

Employing leave-one-out cross-validation on both the OBS and ACT random forests to classify all patients as either *CHEMO-REC*, *CHEMO-NOTREC*, or *INC*, we consider survival differences among these groups as measured by the log-rank test. Of prime interest is the comparison of those patients who received the correct treatment (according to the model prediction), versus those who did not receive the correct treatment. The correct treatment group consists of those OBS patients classified as *CHEMO-NOTREC* and those ACT patients classified as *CHEMO-REC*. The incorrect treatment group consists of *OBS* patients classified as *CHEMO-REC* (meaning they should have taken ACT treatment) and ACT patients classified as *CHEMO-NOTREC*

(meaning they should not have taken ACT treatment). These groups show a significant overall survival difference (log rank $P$=0.0002, $\chi^2$ = 13.856; FIGURE 1), with the correct treatment group showing much better survival times than the incorrect treatment group. The log rank test also shows significant survival differences in other comparisons of interest. Comparing ACT patients predicted as *CHEMO-REC* to ACT patients predicted as *CHEMO-NOTREC* shows a significant survival difference (log rank $P$=0.0012, $\chi^2$ = 10.4; FIGURE 8), with ACT *CHEMO-REC* patients having better survival. This indicates the model is effective at distinguishing patients who will receive benefit from chemotherapy treatment from those who will not.

OBS patients classified by the model as *CHEMO-NOTREC* compared with OBS patients classified as *CHEMO-REC* shows a significant survival difference (log rank $P$=0.0052, $\chi^2$ = 7.8; FIGURE 9), with OBS *CHEMO-NOTREC* patients exhibiting greater overall survival for the duration of the study. This suggests the model is able to predict the prognosis of a patient under surgery alone, with no chemotherapy treatment. Comparing ACT patients classified as *CHEMO-REC* to OBS patients classified as *CHEMO-REC* also showed significantly different survivals (log rank $P$=0.0015, $\chi^2$ = 10.1; FIGURE 10), with the overall survival of OBS *CHEMO-REC* patients markedly lower than that of ACT *CHEMO-REC* patients. ACT *CHEMO-NOTREC* patients had significantly lower survival compared to that of OBS *CHEMO-NOTREC* (log rank $P$=0.0097, $\chi^2$ = 6.7; FIGURE 11). It is of note that the ACT *CHEMO-NOTREC* group shows a steep drop in survival after approximately 5 years, which may be due to late toxicities associated with cisplatin-based chemotherapy (Fossa et al. 2007).

TABLE 2. Summary of LOOCV Log-rank Comparisons

| Comparison | Log-rank Test *P*-values | $\chi^2$ |
|---|---|---|
| Right TX vs Wrong TX | 0.0002 | 13.856 |
| ACT *CHEMO-REC* vs ACT *CHEMO-NOTREC* | 0.0012 | 10.4 |
| OBS *CHEMO-REC* vs OBS *CHEMO-NOTREC* | 0.0052 | 7.8 |
| ACT *CHEMO-REC* vs OBS *CHEMO-REC* | 0.0015 | 10.1 |
| ACT *CHEMO-NOTREC* vs OBS *CHEMO-NOTREC* | 0.0097 | 6.7 |

5.1.2 Predictive Performance Measures

The confusion matrix resulting from the cross-validation procedure is given in TABLE 1. The estimated performance of the classification algorithm is accuracy 68.8%, sensitivity 62.9%, specificity 75.9%, and false-discovery rate 24.1%. Of the 133 patients classified by the model, 29 were classified as *CHEMO-REC*, 35 were classified as *CHEMO-NOTREC*, and 69 patients produced inconclusive results. These results show the model exhibits significantly greater accuracy in identifying patients who should not take chemotherapy, and so is conservative with respect to the recommendation of chemotherapy treatment. This is in addition to inherent conservatism built into the model through returning inconclusive results for many patients. As a consequence, the model has a relatively high positive predictive value (PPV) at 75.9%. For the purposes of

clinical applications, this model behavior is preferred over a more aggressive approach due to the toxicity of the chemotherapy treatment. The application of this classification algorithm could potentially allow healthcare practitioners to more selectively apply ACT treatment and thereby reduce deaths due to treatment toxicity.

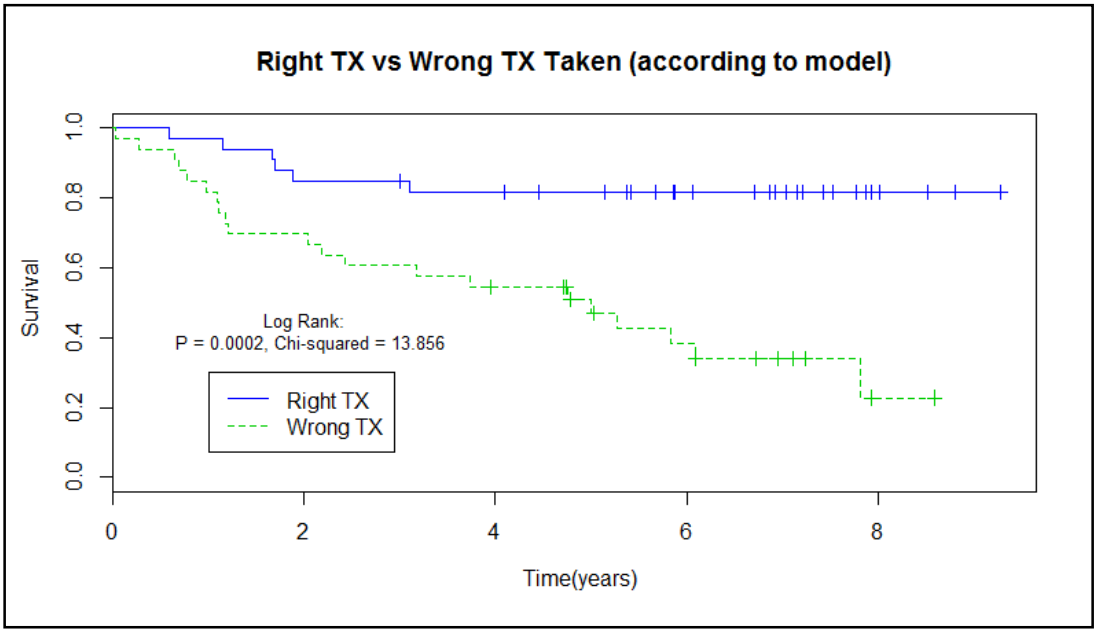


FIGURE 1. Survival curves for Right Treatment vs Wrong Treatment groups as predicted by the model.

TABLE 3. Cross-Validation Confusion Matrix, Single Run of the Model.

| | True *CHEMO-REC* | True *CHEMO-NOTREC* | |
|---|---|---|---|
| Predicted *CHEMO-REC* | 22 | 7 | PPV: 0.7586 |
| Predicted *CHEMO-NOTREC* | 13 | 22 | NPV: 0.6286 |
| | TPR: 0.6286 | TNR: 0.7586 | ACC: 0.6875 |
| | FNR: 0.3714 | FPR: 0.24 | |

5.1.3 Subgroup Analysis by Stage

Standard treatment options for Stage I and Stage II NSCLC patients include surgery and chemotherapy following surgery (National Cancer Institute 2015). Statistical analysis of the JBR.10 trial showed a significant positive benefit of chemotherapy for Stage II patients, but the degree of benefit, if any, for stage I patients remains unclear (Winton et al. 2005). Considering the patient population stratified by stages, a key issue in improving NSCLC treatment is the identification of subgroups within a stage that exhibit a higher likelihood of benefit from ACT treatment. The 15 gene signature derived by Zhu et al. showed that high-risk (as determined by the gene signature) stage IB patients did receive survival benefit from ACT patients, while low-risk patients showed a possibly detrimental effect as a result of chemotherapy (Zhu et al. 2010). Stratifying the patients by stage, we compare the survival of patients who received the correct treatment (as predicted by the model) to those who received the incorrect treatment. The model classification of stage I patients from the JBR.10 data set showed

that patients who received the correct treatment exhibited significantly prolonged survival (log rank $P$=0.0335, $\chi^2$ = 4.5; FIGURE 2).

Stage II patients showed survival benefit among the correct treatment subgroup (log rank $P$=0.0073, $\chi^2$ = 7.2; FIGURE 3). The model offers an alternative categorization of patients relative to cancer stage, and may thus be of use in the development of personalized treatments for patients based on their gene expression levels.

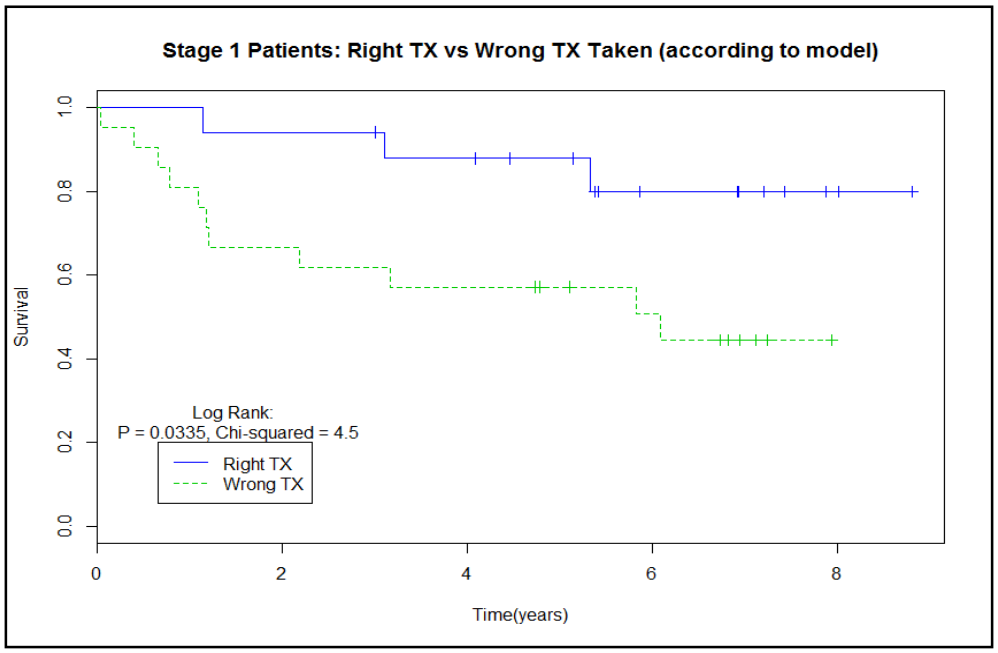


FIGURE 2. Survival curves for Right Treatment vs Wrong Treatment for Stage I patients.

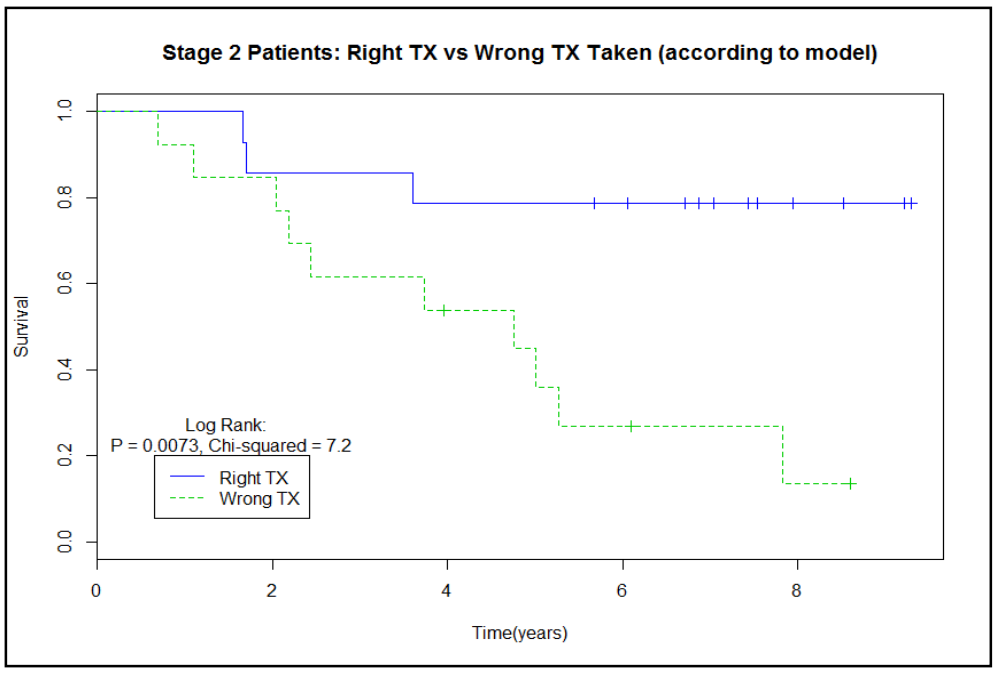


FIGURE 3. Survival curves for Right Treatment vs Wrong Treatment for Stage II patients.

## 5.2 Average Model Performance

Due to the random nature of the random forest model, which relies on bootstrap aggregation of survival trees, each instance of the model will have different performance measures. Random forests have been shown to have relatively low variance, so we expect each individual instance to provide a good indication of overall performance. To better understand the approximate average model performance and variance of the performance measures, we constructed 100 instances of the random forest classification model and calculated performance measures for each instance.

Summary numbers are provided in Tables 3-4. The boxplots of log-rank chi-squared values are given in FIGURE 4; boxplots of accuracy measures are provided in FIGURE 5. The boxplots and means indicate that the classifications produced by the

model consistently have significantly different survival characteristics with an acceptable amount of variation across the model instances. The critical value for 95% significance under a log-rank test with one degree of freedom is 3.8415. We consider the five comparisons of patient classifications as above.

The Right TX vs Wrong TX prediction groups have survival curves that are well separated at a high level of significance (average log rank $P$=0.0002, $\chi^2$ = 13.856). From the 100 instances constructed, the minimum log-rank $P$-value was 0.0032. Thus the models produced by this scheme consistently classify patients in a way that yields a significant separation of survival curves for those patients who adopted the correct treatment according to the model compared to those patients who did not adopt the correct treatment according to the model. This is the most important comparison considered as it evaluates the overall effectiveness of the model in recommending the treatment that will produce the greatest survival benefit for the patient.

Other group comparisons yield significant survival separation on average, but at a weaker level than the overall Right TX vs Wrong TX group. Of the four subgroup comparisons, two were found to have non-significant differences for some of the computed model instances: ACT-*REC* vs ACT-*NOTREC* had equivalent survival for 4% of models; ACT-*NOTREC* vs OBS-*NOTREC* had equivalent survival for 25% of models. The two remaining comparisons showed significantly different survival for all computed models, with minimum $P$-values $P$=0.0337 for ACT-*REC* vs OBS-*REC*, and $P$=0.0358 for OBS-*REC* vs OBS-*NOTREC*.

The poorest performance was for the ACT *NOTREC* patients against the OBS *NOTREC* patients. This is a comparison of patients who took chemotherapy but should not have (according to the model) to patients who did not take chemotherapy and should not have. The difficulty of the model to produce good separation of survival curves in this case may be expected, and is possibly due to the different average survival of ACT patients compared to OBS patients, which is reflected in the different cutoff values for classifying ACT patients and OBS patients for the *a priori REC* and *NOTREC* categories. One possible interpretation here is that the survivability of patients who will not benefit from chemotherapy and opt for surgery only is still only marginally better than those patients who receive chemotherapy but are not expected to have survival benefit according to the model.

For the subgroup comparisons, the model performs about equally well for ACT-*REC* vs OBS-*REC*, and for OBS-*REC* vs OBS-*NOTREC*. Good performance for the former comparison indicates that the model is effective at identifying patients who have good survivability under ACT treatment and poor survivability under surgery alone. Good performance in the latter category shows the model's effectiveness in distinguishing patients who have good survivability under surgery alone from those who have poor survivability under surgery alone. This latter subgroup comparison is of note since the ability to identify high-risk patients may be of clinical use in its own right, apart from considerations of chemotherapy treatment.

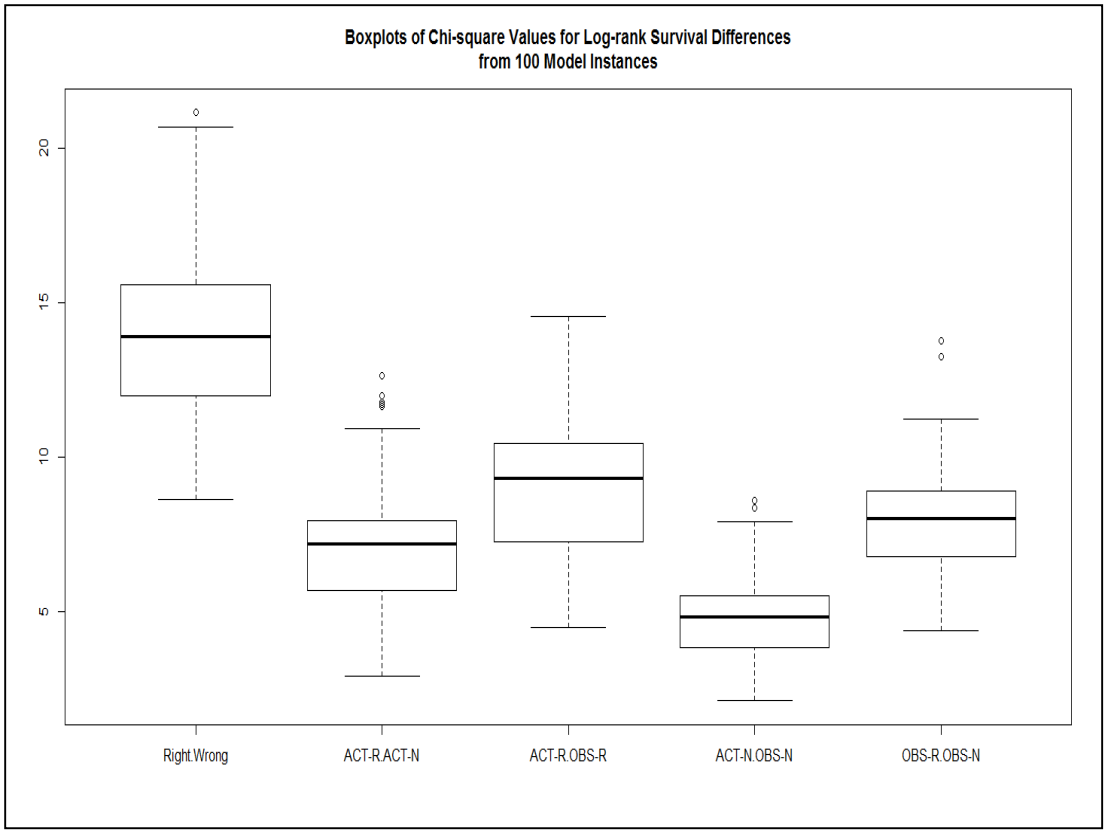


FIGURE 4. Boxplots of log-rank chi-square values for 100 model instances.

TABLE 4. Mean and Standard Deviation for Log-rank Chi-square Test Statistics from 100 Model Instances

| | Mean (P-value) | Standard Deviation |
|---|---|---|
| Right TX vs. Wrong TX | 13.8560 (0.0002) | 2.6113 |
| ACT-*REC* vs ACT-*NOTREC* | 7.121 (0.0076) | 2.0408 |
| ACT-*REC* vs OBS-REC | 9.1007 (0.0026) | 2.1969 |
| ACT-*NOTREC* vs OBS-*NOTREC* | 4.8305 (0.0280) | 1.2462 |
| OBS-*REC* vs OBS-*NOTREC* | 8.0024 (0.0047) | 1.7164 |

TABLE 5. Average Log-rank Chi-square (P-values) for 100 Model Instances

| | Min. | 1st Quartile | Median | 3rd Quartile | Max. |
|---|---|---|---|---|---|
| Right TX vs. Wrong TX | 8.6241 (0.0032) | 11.9888 (0.0005) | 13.9083 (0.0002) | 15.5828 (<0.0001) | 20.6592 (<0.0001) |
| ACT-*REC* vs ACT-*NOTREC* | 2.9261 (0.0872) | 5.7144 (0.0168) | 7.2151 (0.0072) | 7.9535 (0.0048) | 10.9429 (0.0009) |
| ACT-*REC* vs OBS-REC | 4.5112 (0.0337) | 7.2744 (0.0070) | 9.3246 (0.0023) | 10.4465 (0.0012) | 14.5416 (0.0001) |
| ACT-*NOTREC* vs OBS-*NOTREC* | 2.1312 (0.1443) | 3.8413 (0.0500) | 4.8484 (0.0277) | 5.5162 (0.0188) | 7.9334 (0.0049) |
| OBS-*REC* vs OBS-*NOTREC* | 4.4067 (0.0358) | 6.7974 (0.0091) | 8.0135 (0.0046) | 8.9032 (0.0028) | 11.2206 (0.0008) |

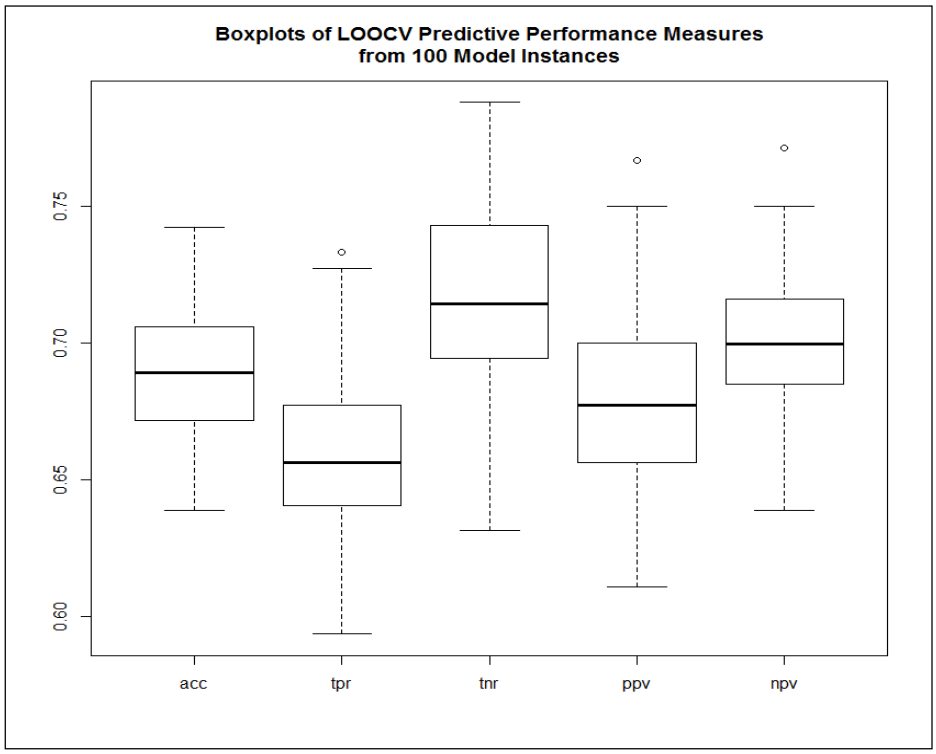


FIGURE 5. Boxplots of LOOCV model accuracies. Accuracy: acc; true positive rate: tpr; true negative rate: tnr; positive predictive value: ppv; negative predictive value: npv.

TABLE 6. Mean and Standard Deviation for Log-rank Chi-square Test Statistics from 100 Model Instances.

| | Mean (P-value) | Standard Deviation |
|---|---|---|
| Accuracy | 0.6899 | 0.0217 |
| TPR | 0.6612 | 0.0285 |
| TNR | 0.7161 | 0.0293 |
| PPV | 0.6787 | 0.0287 |
| NPV | 0.7000 | 0.0254 |

TABLE 7. Average Predictive Performance Over 100 Model Instances.

| | Min. | 1st Quartile | Median | 3rd Quartile | Max. |
|---|---|---|---|---|---|
| Accuracy | 0.6389 | 0.6716 | 0.6893 | 0.7059 | 0.7424 |
| TPR | 0.5938 | 0.6407 | 0.6562 | 0.6774 | 0.7272 |
| TNR | 0.6316 | 0.6944 | 0.7143 | 0.7429 | 0.7879 |
| PPV | 0.6111 | 0.6563 | 0.6774 | 0.7000 | 0.7500 |
| NPV | 0.6389 | 0.6850 | 0.6998 | 0.7161 | 0.7500 |

## 5.3 Estimate of Enhanced Treatment Effect Subgroup

To apply the virtual twins regression method described by Foster, Taylor, and Ruberg (2011), we estimate the treatment $Z_i$ effect for patient i as $Z_i = \widehat{P}_{1i} - \widehat{P}_{0i}$, where $\widehat{P}_{1i}$ is the probability of early death for patient i under ACT treatment, and $\widehat{P}_{0i}$ is the probability of early death under surgery alone. The probabilities $\widehat{P}_{1i}$ and $\widehat{P}_{0i}$ are calculated by applying both random forests to each patient and assigning patient i probabilities according to the percentage of trees that classify a patient as either REC or NOTREC respectively.

These $Z_i$ are used as the response variable in a regression tree trained on the union of all ACT and OBS patients (FIGURE 6***)***. From the regression tree, we identify the region of enhanced treatment effect as defined by those terminal nodes with predicted

treatment effect less than the threshold $c = \delta - 0.05$ = -0.1113 where $\delta = -0.0613$ is the estimated treatment effect for the patient population. In FIGURE 6, the nodes with estimated treatment effects $-0.46, -0.19, -0.33, -0.27$ define the estimated enhanced treatment effect region. In terms of probe set expression values, the estimated region $\hat{A}$ contains patients whose data satisfy one of the following criteria: i) $(X214798\_at < -0.44) \cap (X217100\_s\_at < 0.78)$; ii) $(X214798\_at \geq -0.44) \cap (X218295\_s\_at < -0.44) \cap (X214146\_s\_at < 0.13)$; iii) $(X214798\_at \geq -0.44) \cap (X218295\_s\_at \geq -0.44) \cap (X203998\_s\_at \geq 0.15)$.

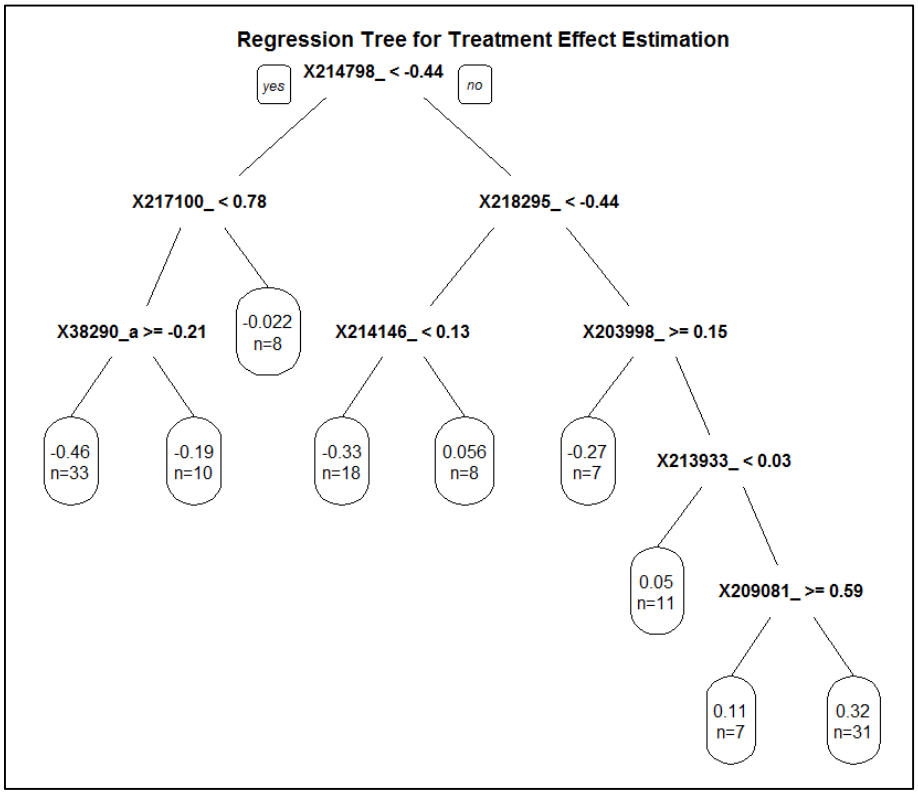


FIGURE 6. Regression tree for treatment effect estimation.

Applying the bias corrected bootstrap for the region $\hat{A}$ described above, we calculate the treatment effect for the enhanced effect region to be -0.3583. This can be interpreted as a 35.83% reduction in likelihood of early death under ACT treatment compared to no ACT treatment. Thus the enhanced treatment effect region defines a subgroup of the patient population that has a higher predicted likelihood of survival benefit from chemotherapy treatment, independent of cancer stage.

## 5.4 External Validation with an Independent Test Set

### 5.4.1 Test Set Data Description

The UHN181 independent validation set generated for a follow-up validation of the 15 gene prognostic signature found by Zhu et al. (2010) includes 181 early stage non-small-cell lung cancer patients who underwent surgery alone, and did not receive ACT therapy (Der et al. 2014). For this data set, tumor samples were obtained from NSCLC patients who underwent surgical resection at University Health Network during 1996-2005. The snap frozen tumor samples were retrieved from the UHN tumor bank, and total RNA was extracted following the same methods of the original study. Processing was performed with Affymetrix U133 2.0 Plus arrays. The resulting microarray data were normalized using RMA and deposited with GEO (GSE50081).

Although the JBR.10 cohort contains only stage IB and II patients, the UHN181 cohort contains stage IA, IB, and stage II patients. JBR.10 is composed of 55% stage IB and 45% stage II patients, while UHN181 is 71% stage I and 29% stage II patients. A comparison of the demographics of UHN181 and the training data set JBR.10 are given in TABLE 8. The processed UHN181 data for this study were retrieved from GEO.

After centering and scaling, we applied the classification algorithm to the test data and considered the resulting classification groups for 5-year survival difference, following the validation procedure of Der et al. (2014).

TABLE 8. Comparison of Clinical Characteristics for JBR.10 and UHN181.

| Variable | JBR.10 (n=62) | UHN181 (n=181) |
|---|---|---|
| Age | | |
| Median age (range) | 61 (35-77) | 69.7 (40.2-87.9) |
| <65 years | 43 (69%) | 59 (33%) |
| >=65 years | 19 (31%) | 122 (67%) |
| Sex | | |
| Female | 18 (29%) | 83 (46%) |
| Male | 44 (71%) | 98 (54%) |
| Histology | | |
| Adenocarcinoma | 32 (52%) | 128 (71%) |
| Squamous cell carcinoma | 26 (42%) | 43 (24%) |
| Others | 4 (6%) | 10 (6%) |
| Stage | | |
| IA | 0 | 48 (27%) |
| IB | 34 (55%) | 79 (44%) |
| IIA | 28 (45%) | 9 (5%) |
| IIB | 0 | 45 (25%) |

5.4.2 Test Set Results

The OBS random forest algorithm separates the independent test set into subgroups *CHEMO-REC* (n=89) and *CHEMO-NOTREC* (n=92), but these subgroups did not exhibit a significant 5-year survival difference ($P$=0.125). Kaplan-Meier survival

curves for each subgroup are given in Figure 9. We see that the two curves cross twice prior to two years, but that overall the *CHEMO-NOTREC* group appears to have higher 5-year survival, which supports the performance of the model. The accuracy of the OBS random forest applied to the test data is 52.5%, much lower than the accuracy estimated via cross-validation on the training set of 68.99%.

Due to a lack of suitable studies providing genetic data for an independent set of ACT patients, we were unable to verify the ACT random forest in a similar manner.

## 5.5 Identification of Important Probe Sets

The random forest classifications permit a measure of covariate importance through mean decrease in accuracy and mean decrease in the Gini coefficient. Plots of the most important probe sets by these measures are given in Figures 13 and 14, and identified in Tables 9 and 10. The mean decrease in accuracy for each probe is calculated by comparing the predictive performance of the original forest on the out-of-bag data to the predictive performance after permuting each predictor. The second measure is calculated as the mean decrease in Gini coefficient from splitting on the predictor, averaged over all trees in the random forest.

The functions of these identified important probe sets listed in the tables show many potential links with cell growth and cell regulation. Considering the three most important probe sets from the OBS random forest, NUP107 is related to activity of the cell nuclear core, DSCC1 is associated with DNA replication, and YEATS4 has been shown to have amplified expression in tumors. The most important probe set from the ACT random forest is MPP2, which produces a mean decrease in accuracy of 14.407,

much larger than any other probe set in either of the forests. MPP2 is responsible for regulate cell replication and signaling pathways within the cell (Weizman Institue of Science 2012).

Table 11 identifies the name and some basic function of the five gene signature that forms the VTR estimated enhanced treatment effect region. These probe sets are disjoint from those identified by the random forests, but there is some similarity in cell functions in many of the identified genes. The most important gene from the VTR tree is identified by the root node split on probe set 213270_at, corresponding to gene ATP2C2. This gene is related to ATP processing, and along with SYT1, is involved in calcium transport and binding. NUP50 and SYT1 are membrane proteins responsible for vesicular transport and protein import across the cell membrane. NUP50 and PPBP play a role in cell proliferation, with PPBP also related to DNA synthesis, mitosis, along with a number of other functions (Weizman Institue of Science 2012).

Future work could consider the function of these genes in more detail as they relate to tumor growth and patient response to chemotherapy. These genes are disjoint from those identified in the 15-gene signature of Zhu (as is common with gene signature studies), but it may be that these genes serve similar roles or are related in functioning to the genes found in previous gene signatures; future clinical work could be done to develop a better understanding of how the disjoint gene signatures found in different studies are related.

TABLE 9. Important Probe Sets, OBS Random Forest

| Probe Set ID | Mean Decrease Accuracy | Mean Decrease Gini | Gene Symbol | Function |
|---|---|---|---|---|
| 218768_at | 7.906 | 0.922 | NUP107 | Component of nuclear core complex. |
| 219000_s_at | 7.594 | 2.016 | DSCC1 | DNA replication and sister chromatid cohesion 1. |
| 218911_at | 5.021 | 0.945 | YEATS4 | Thought to be required for RNA transcription, amplified in tumors. |
| 214829_at | 3.706 | 0.372 | AASS | |
| 203147_s_at | 3.546 | 0.708 | TRIM14 | |
| 212663_at | 3.544 | 0.336 | FKBP15 | |
| 201947_s_at | 3.311 | 0.515 | CCT2 | Molecular chaperone, assists in folding of proteins. |
| 220076_at | 3.019 | 0.750 | ANKH | Regulates intra and extraceullar levels of inorganic compounds. |
| 212528_at | 2.996 | 0.769 | DESl1 | |

TABLE 10: Important Probe Sets, ACT Random Forest

| Probe Set ID | Mean Decrease Accuracy | Mean Decrease Gini | Gene Symbol | Function |
|---|---|---|---|---|
| 213270_at | 14.407 | 3.286 | MPP2 | Regulates cell proliferation, signaling pathways. |
| 210300_at | 8.674 | 1.927 | REM1 | Promotes endothelial cell sprouting. |
| 220406_at | 7.917 | 1.352 | SNPH | |
| 205104_at | 6.415 | 1.552 | TGFB2 | Transforming growth factor. |
| 203730_s_at | 4.763 | 1.016 | ZKSCAN5 | Zinc finger protein, DNA binding transcription factor activity. |
| 217544_at | 4.745 | 0.622 | LOC729806 | |

TABLE 11: Important Probe Sets, VTR Enhanced Treatment Region

| Probe Set ID | Gene Symbol | Function |
|---|---|---|
| 214798_at | ATP2C2 | Protein coding, calcium transport. |
| 217100_s_at | UBXN7 | Protein coding, transcription factor binding. |
| 218295_s_at | NUP50 | Nucleoporin 50kDa, nuclear pore complex protein related to protein import. Related to CDKN1B, which controls cell proliferation. |
| 214146_s_at | PPBP | Platelet-derived growth factor, stimulates DNA synthesis, mitosis, and glycolysis. |
| 203998_s_at | SYT1 | Membrane protein, related to calcium binding and synapse triggering for vesicular processing. |

# CHAPTER 6

# CONCLUSION

The main difficulty in identifying genes that are predictive of benefit from ACT treatment is validation of the performance of the prediction algorithm, particularly in the absence of independent validation sets from randomized controlled experiments similar to the study used to generate the JBR.10 data set. In lieu of an independent ACT data set, we have used leave-one-out cross-validation to estimate the predictive performance of the classification scheme on independent data. LOOCV estimators have been shown to have low bias, but potentially high variance (Tibshirani, Friedman, and Hastie 2009, 243). The LOOCV estimates here show very good performance of the classifiers, suggesting that the classification scheme may be effective, despite potentially high variance of the LOOCV estimators.

In classifying microarray data, random forests have been shown to produce good results, often with lower variance than competing methods, such as a classification tree. This lower variance may be of particular importance in these studies owing to the common difficulty of reproducing and verify the identified gene signature. For the present model design, the analysis of average model performance gives a strong indication that the model variability is acceptably small, such that multiple model instances produce approximately similar results with respect to predictive accuracy and

separation of survival characteristics for relevant subgroup comparisons. Even with the worst performing comparison for ACT-*NOTREC* vs OBS-*NOTREC*, 75% of the model instances produced classifications with significantly different survival characteristics for these subgroups. The most important comparison, Right TX vs Wrong TX, showed the strongest results, with a minimum *P*-value of $P=0.0033$ for the sample instances constructed, thus the model appears to be effective in recommending the treatment for a patient that will lead to greatest survival benefit. The trade-off with the random forest is an increase in bias and a decreased interpretability in the relationships of significant genes. We can still achieve some idea of which genes are most significant by measuring the mean decrease in accuracy and mean decrease in Gini coefficient for the probe sets. The regression tree calculated in the second stage of the model provides a further identification of a significant set of genes through the estimated enhanced treatment effect region, which is defined by expression levels on six probe sets only. Using the bias corrected bootstrap method, the estimated treatment effect for the enhanced region is -0.3583, which can be interpreted as the reduction in probability of early death under ACT treatment over no ACT treatment.

While the cross-validation results supported the effectiveness of the proposed classification model, the results of applying the algorithm to the test data were fairly weak. Even though it is anticipated, it is not clear what may be the reason for the large decrease in predictive accuracy and overall nonsignificant separation of survival curves for the classification subgroups. One possible cause may be batch effect error in comparing microarray data from separate studies; however, the validation data set was

generated via the same methods as the training set specifically for validation of the gene signature based on the JBR.10 data (Zhu et al. 2010; Der et al. 2014), so this source of error may be unlikely. A future analysis of the data could consider processing the training and test data sets to remove possible batch effects, as suggested by (Luo et al. 2010).

The results of the present study largely corroborate the findings of Zhu et al. (2010) and provide an alternative approach to the identification of a gene signature that is predictive of survival benefit from ACT treatment. These results can be used to develop targeted therapies for patients based on their gene expression levels and their likelihood of survival benefit from chemotherapy treatment. While ACT treatment is the standard of care for stage II patients, the use of ACT for stage I patients remains controversial (Zhu et al. 2010). Also in accordance with the results from Zhu et al. (2010), the predictive model presented here shows the plausibility of adopting different treatment strategies for stage I patients based on their likelihood of survival benefit as predicted by their microarray data. This is of particular importance since chemotherapy treatment clearly shows an overall survival benefit in the JBR.10 data set, but, as shown by Zhu et al. (2010), low-risk stage I patients may actually have decreased survival from ACT treatment.

The possibility of developing personalized treatment plans for early-stage NSCLC patients from microarray data using a random forest based classification appears to be feasible. In order to develop the results here into a practical clinical method, more validation is needed, ideally with independent data sets that result from similar trials and

laboratory analysis as the JBR.10 study. A more thorough analysis of the characteristics of the test data set UHN181 with respect to the training data set may be warranted to determine possible causes for the poorer performance with the test data. As more data from appropriate randomized clinical trials comparing ACT treatment with an observation group become available, the model could be retrained with larger samples to improve the power and accuracy of the resulting predictions. Future work may also consider refinements of the model presented here through a finer use of survival and censoring information with a regression or survival random forest as opposed to a classification random forest; and through modifications of the model parameters, for instance changing the survival cutoffs of classification groups. The model decision scheme could also be adapted to more aggressively prescribe ACT treatment for chemotherapy by classifying the *CHOICE* patients as *CHEOM-REC* instead.

# APPENDICES

# APPENDIX A

# ADDITIONAL FIGURES

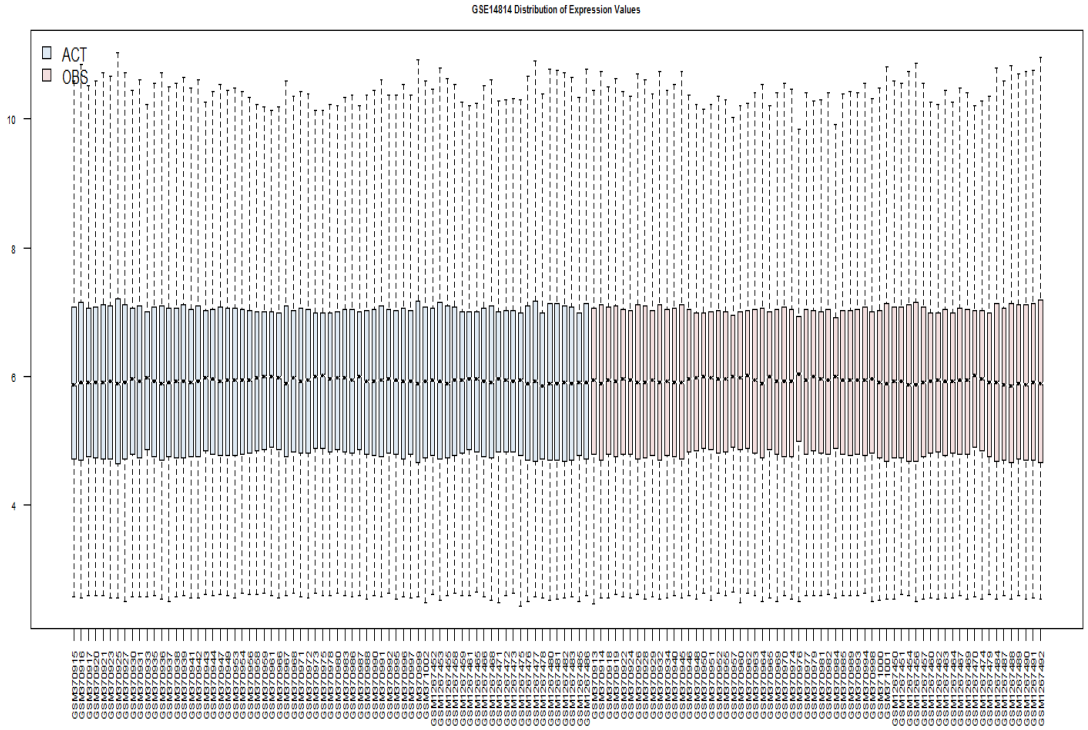


FIGURE 7. Boxplots of probe set expression value distributions by patient.

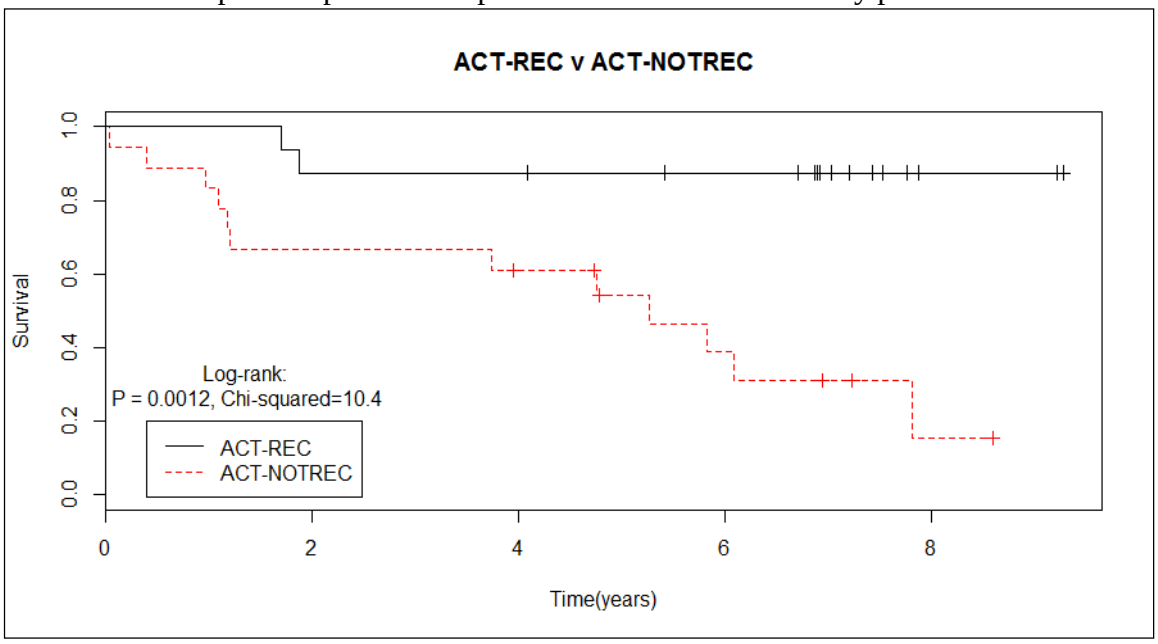


FIGURE 8. Survival curves for ACT-REC vs ACT-NOTREC.

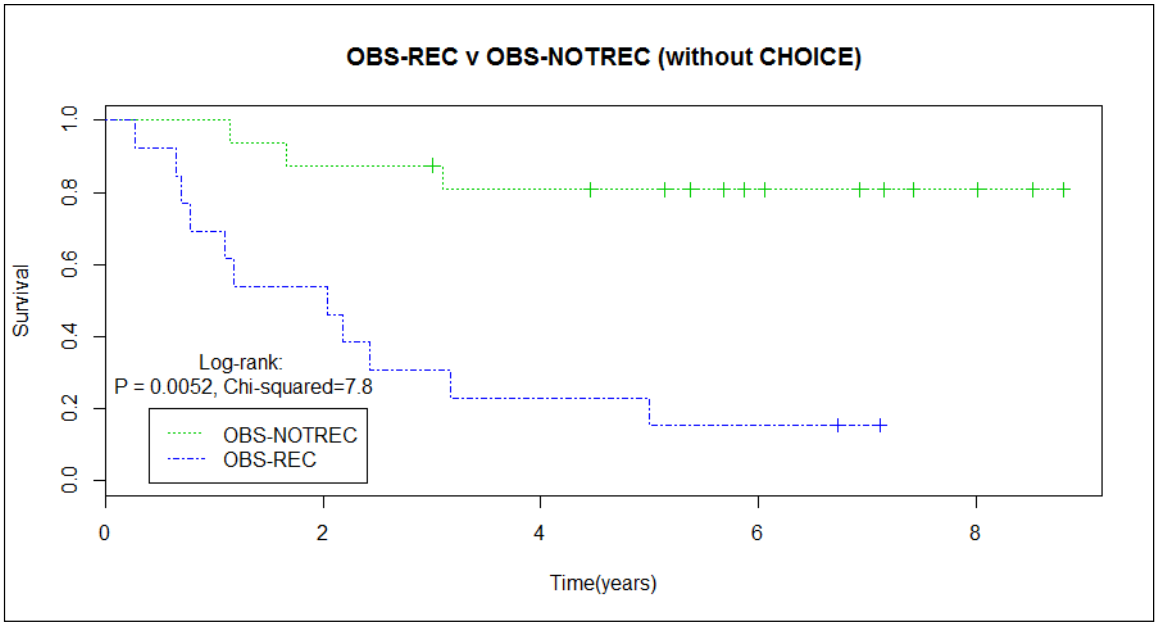


FIGURE 9. Survival curves for OBS-REC vs OBS-NOTREC.

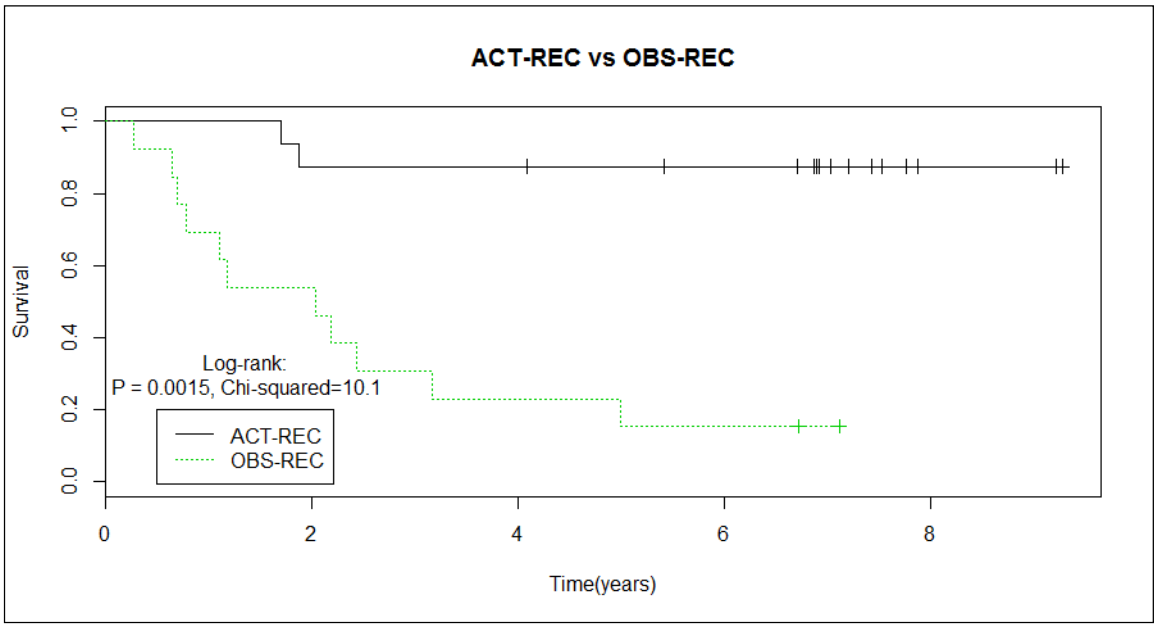


FIGURE 10. Survival curves for ACT-REC vs OBS-REC.

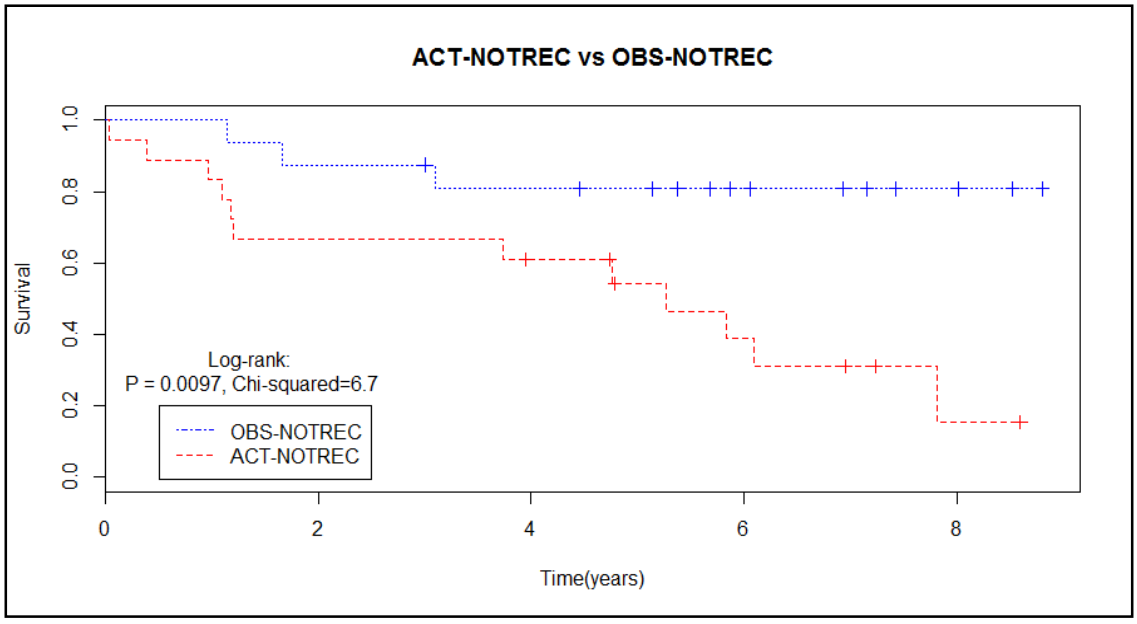


FIGURE 11. Survival curves for ACT-NOTREC vs OBS-NOTREC.

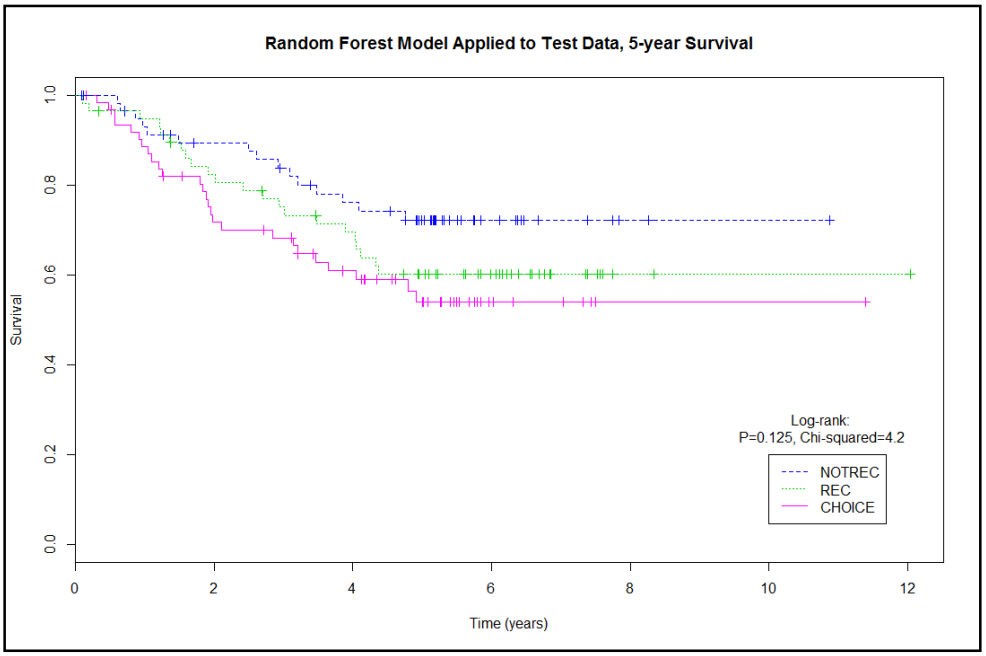


FIGURE 12. Kaplan-Meier survival estimates for model predicted classes on the test data.

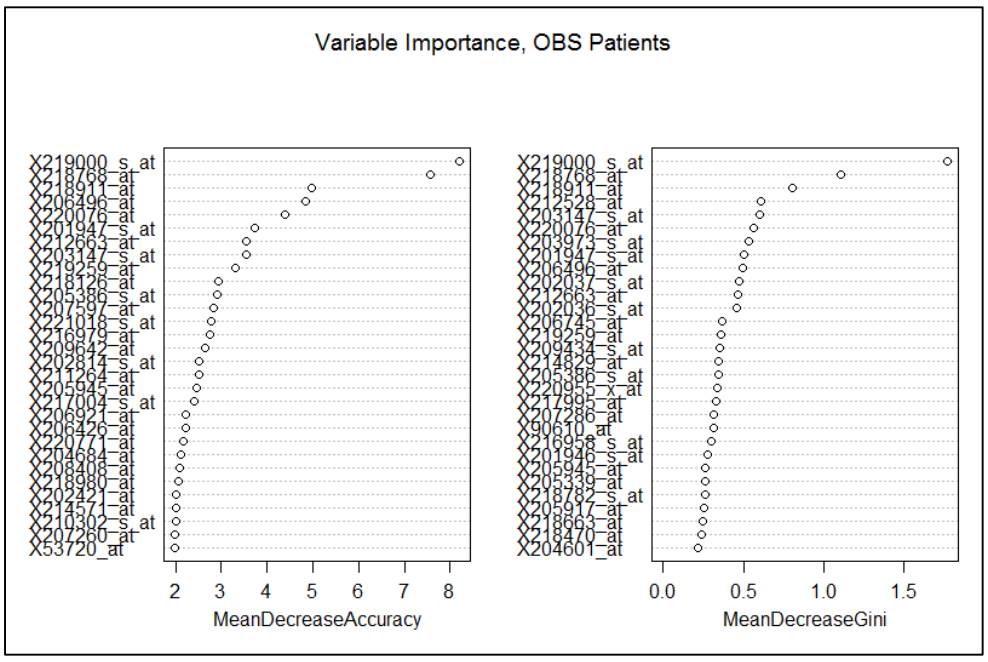


FIGURE 13. Variable importance measures from OBS random forest.

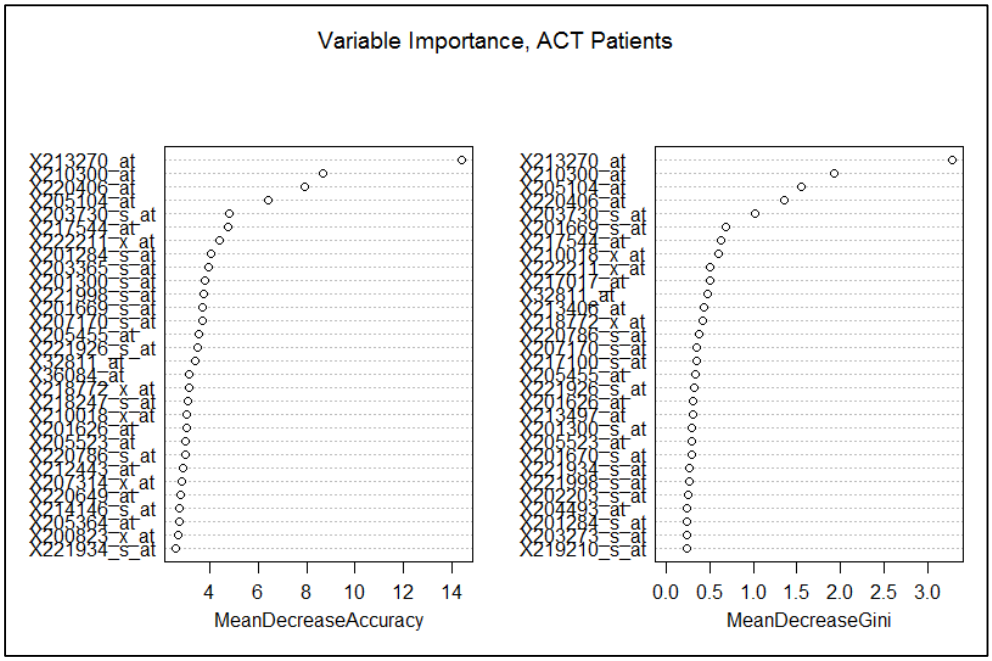


FIGURE 14. Variable importance measures from ACT random forest.

# APPENDIX B

## R CODE

```
library(randomForest)
library(survival)

#################################################################
#                                                               #
#  Functions:                                                   #
#    1. getcvResults(predictors, survival.data)                 #
#                                                               #
#################################################################

getcvResults <- function(predictors, survival.data, ntree.param, mtry.param) {
  predictors <- scale(predictors)
  correct = 0
  misclass = 0
  patient.classes <- rep('', nrow(survival.data))
  names(patient.classes) <- rownames(survival.data)
  for(i in 1:nrow(survival.data)) {
    loocv.predictors <- predictors[-i, ]
    loocv.survival <- survival.data[-i,]
    test.patient.predictors <- predictors[i,]
    loocv.rf.data <- data.frame(loocv.predictors, given.class=loocv.survival$given.class,
OS=loocv.survival$OS)
    loocv.rf <- randomForest(as.formula(paste('given.class ~',
paste(colnames(loocv.predictors), collapse='+'))),
                             data=loocv.rf.data, importance=TRUE, proximity=TRUE,
ntree=ntree.param, mtry=mtry.param)
    loocv.predicted <- predict(loocv.rf, predictors)
    test.patient.predicted <- loocv.predicted[i]

    if (test.patient.predicted == survival.data$given.class[i]) {
      correct = correct + 1
    } else {
      misclass = misclass + 1
    }

    patient.classes[i] <- test.patient.predicted
  }
  return(patient.classes)
```

```
}

cvTuning <- function(obs.scaled, act.scaled, obs.survival, act.survival,
                mtry.start, mtry.end, mtry.inc=10, ntree.start,
                ntree.end, ntree.inc=10) {
  cntr <- 0
  mtry.seq <- seq(mtry.start, mtry.end, mtry.inc)
  ntree.seq <- seq(ntree.start, ntree.end, ntree.inc)
  results <- matrix(, nrow=length(mtry.seq)*length(ntree.seq), ncol=5)
  for(mtry in mtry.seq){
    for(ntree in ntree.seq){
      cntr <- cntr + 1
      obs.correct <- c()
      act.correct <- c()
      obs.choice <- c()
      act.choice <- c()
      act.cv.results <- getcvResults(act.scaled, act.survival, mtry.param=mtry,
                              ntree.param=ntree)
      obs.cv.results <- getcvResults(obs.scaled, obs.survival, mtry.param=mtry,
                              ntree.param=ntree)

      obs.rf.data <- data.frame(obs.scaled,
                            given.class=obs.survival$given.class,
                            OS=obs.survival$OS, stringsAsFactors=TRUE)
      obs.rf <- randomForest(as.formula(paste('given.class ~',
                                      paste(colnames(obs),
                                          collapse='+'))),
                          data=obs.rf.data, importance=TRUE,
                          proximity=TRUE, ntree=1000, mtry=150,
                          norm.votes=TRUE)

      act.rf.data <- data.frame(act.scaled,
                            given.class=act.survival$given.class,
                            OS=act.survival$OS)
      act.rf <- randomForest(as.formula(paste('given.class ~',
                                      paste(colnames(act.scaled),
                                          collapse='+'))),
                          data=act.rf.data, importance=TRUE, proximity=TRUE,
                          ntree=1000, mtry=150)
```

```
act.cv.results <- c(act.cv.results, predict(act.rf,
                                obs.predictors.for.actrf))
obs.cv.results <- c(obs.cv.results, predict(obs.rf,
                                act.predictors.for.obsrf))

reclist <- rep('', 62)
names(reclist) <- rownames(obs.survival)
for (patient in rownames(obs.survival)){
  if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'REC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'NOTREC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'CHOICE'
  } else if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'CHOICE'
  }
}
obs.survival$recommendation <- reclist

reclist <- rep('', 71)
names(reclist) <- rownames(act.survival)
for (patient in rownames(act.survival)){
  if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'REC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'NOTREC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'CHOICE'
  } else if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'CHOICE'
  }
}
act.survival$recommendation <- reclist

for(patient in rownames(act.survival)){
  if(act.survival[patient, 'recommendation']=='CHOICE'){
    act.choice <- c(act.choice, patient)
```

```
      } else if(act.survival[patient, 'recommendation'] == act.survival[patient,
'given.class']){
        act.correct <- c(act.correct, patient)
        if(act.survival[patient, 'given.class']=='REC'){
          rec.correct <- c(rec.correct, patient)
        } else if(act.survival[patient, 'given.class']=='NOTREC') {
          notrec.correct <- c(notrec.correct, patient)
        }
      } else {
        act.misclass <- c(act.misclass, patient)
        if(act.survival[patient, 'given.class']=='REC'){
          rec.misclass <- c(rec.misclass, patient)
        } else if(act.survival[patient, 'given.class']=='NOTREC'){
          notrec.misclass <- c(notrec.misclass, patient)
        }
      }
    }

    for(patient in rownames(obs.survival)){
      if(obs.survival[patient, 'recommendation']=='CHOICE'){
        obs.choice <- c(obs.choice, patient)
      } else if(obs.survival[patient, 'recommendation'] == obs.survival[patient,
'given.class']){
        obs.correct <- c(obs.correct, patient)
        if(obs.survival[patient, 'given.class']=='REC'){
          rec.correct <- c(rec.correct, patient)
        } else if(obs.survival[patient, 'given.class']=='NOTREC') {
          notrec.correct <- c(notrec.correct, patient)
        }
      } else {
        obs.misclass <- c(obs.misclass, patient)
        if(obs.survival[patient, 'given.class']=='REC'){
          rec.misclass <- c(rec.misclass, patient)
        } else if(obs.survival[patient, 'given.class']=='NOTREC'){
          notrec.misclass <- c(notrec.misclass, patient)
        }
      }
    }
```

```
    results[cntr, 1] <- (length(act.correct)+length(obs.correct))/(133-length(act.choice)-
length(obs.choice))
    results[cntr, 2] <- length(act.correct)+length(obs.correct)
    results[cntr, 3] <- length(act.choice)+length(obs.choice)
    results[cntr, 4] <- mtry
    results[cntr, 5] <- ntree
    print(results[cntr,])
   }
  }
 return(results)
}

cvTuning.results <- cvTuning(obs.scaled, act.scaled, obs.survival,
                        act.survival, 25, 25, 0, 400, 2000, 100)

##########################################
#                                        #
#  Data input and setup                  #
#                                        #
##########################################

#bothTX <- read.csv('Data/bothTX_all_predictors.csv', row.names=1)
act <- read.csv('Data/act_selected_predictors.csv', row.names=1)
obs <- read.csv('Data/obs_selected_predictors.csv', row.names=1)
act.survival <- read.csv('Data/gse14814_act_survival.csv', stringsAsFactors=F,
colClasses=c('character', 'character', 'character', 'numeric', 'character', 'character',
'character', 'numeric', 'character', 'numeric'), row.names=1)
obs.survival <- read.csv('Data/gse14814_obs_survival.csv', stringsAsFactors=F,
colClasses=c('character', 'character', 'character', 'numeric', 'character', 'character',
'character', 'numeric', 'character', 'numeric'), row.names=1)
act.scaled <- scale(act)
obs.scaled <- scale(obs)

act.predictors.for.obsrf <- scale(bothTX[rownames(act), colnames(obs)])
obs.predictors.for.actrf <- scale(bothTX[rownames(obs), colnames(act)])

test.obs.predictors <- read.csv('Data/gse50081_obs_predictors.csv', row.names=1,
colClasses=c('character', rep('numeric', 496)), header=T)
test.obs.scaled <- scale(test.obs.predictors)
```

```
test.act.predictors <- read.csv('Data/gse50081_act_predictors.csv', row.names=1,
colClasses=c('character', rep('numeric', 406)), header=T)
test.act.scaled <- scale(test.act.predictors)
test.survival <- read.csv('Data/gse50081_survival.csv', row.names=1, header=T,
stringsAsFactors=F, colClasses=c(rep('character', 4), rep('numeric', 3)))

obs.survival.cutoff <- 3.815
obs.survival$given.class[obs.survival$OS > obs.survival.cutoff] <- 'NOTREC'
obs.survival$given.class[obs.survival$OS <= obs.survival.cutoff] <- 'REC'
act.survival.cutoff <- 5.81
act.survival$given.class[act.survival$OS > act.survival.cutoff] <- 'REC'
act.survival$given.class[act.survival$OS <= act.survival.cutoff] <- 'NOTREC'

########################################
#                                      #
#         CV Tuning                    #
#                                      #
########################################

cv.tuning.results <- cvTuning(obs.scaled, act.scaled, obs.survival,
                      act.survival, mtry.start=100, mtry.end=101,
                      mtry.inc=1, ntree.start=1000, ntree.end=1001,
                      ntree.inc=1)
########################################
#                                      #
#  Make and view the OBS.RF            #
#                                      #
########################################

obs.rf.data <- data.frame(obs.scaled, given.class=obs.survival$given.class,
                    OS=obs.survival$OS, stringsAsFactors=TRUE)
obs.rf <- randomForest(as.formula(paste('given.class ~', paste(colnames(obs),
                collapse='+'))), data=obs.rf.data, importance=TRUE,
                proximity=TRUE, ntree=500, mtry=25, norm.votes=TRUE)
obs.rf

########################################
#                                      #
#  Make and view the ACT.RF            #
```

```
#                                  #
########################################

act.rf.data <- data.frame(act.scaled, given.class=act.survival$given.class,
OS=act.survival$OS)
act.rf <- randomForest(as.formula(paste('given.class ~', paste(colnames(act.scaled),
collapse='+'))),
                       data=act.rf.data, importance=TRUE, proximity=TRUE, ntree=500,
mtry=25)
act.rf

########################################
#                                  #
#  Validate OBS.RF                 #
#    and ACT.RF via LOOCV          #
#                                  #
########################################

#1=NOTREC, 2=REC
act.cv.results <- getcvResults(act.scaled, act.survival, ntree.param=500, mtry.param=25)
obs.cv.results <- getcvResults(obs.scaled, obs.survival, ntree.param=500,
mtry.param=25)

act.cv.results <- c(act.cv.results, predict(act.rf, obs.predictors.for.actrf))
obs.cv.results <- c(obs.cv.results, predict(obs.rf, act.predictors.for.obsrf))

reclist <- rep('', 62)
names(reclist) <- rownames(obs.survival)
for (patient in rownames(obs.survival)){
  if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'REC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'NOTREC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'CHOICE'
  } else if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'CHOICE'
  }
}
```

```
obs.survival$recommendation <- reclist

reclist <- rep('', 71)
names(reclist) <- rownames(act.survival)
for (patient in rownames(act.survival)) {
  if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'REC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'NOTREC'
  } else if (act.cv.results[patient] == '1' & obs.cv.results[patient] == '2') {
    reclist[patient] <- 'CHOICE'
  } else if (act.cv.results[patient] == '2' & obs.cv.results[patient] == '1') {
    reclist[patient] <- 'CHOICE'
  }
}
act.survival$recommendation <- reclist

act.fit.notrec <- survfit(Surv(OS, OSCens)~recommendation,
data=act.survival[act.survival$recommendation=='NOTREC',], conf.int=F)
act.fit.rec <- survfit(Surv(OS, OSCens)~recommendation,
data=act.survival[act.survival$recommendation=='REC',], conf.int=F)

obs.fit.rec <- survfit(Surv(OS, OSCens)~recommendation,
data=obs.survival[obs.survival$recommendation=='REC',], conf.int=F)
obs.fit.notrec <- survfit(Surv(OS, OSCens)~recommendation,
data=obs.survival[obs.survival$recommendation=='NOTREC',], conf.int=F)

rec.survival <- rbind(act.survival[act.survival$recommendation=='REC',],
                      obs.survival[obs.survival$recommendation=='REC',])

notrec.survival <- rbind(act.survival[act.survival$recommendation=='NOTREC',],
                         obs.survival[obs.survival$recommendation=='NOTREC',])


correct.survival <- rbind(act.survival[act.survival$recommendation=='REC',],
                          obs.survival[obs.survival$recommendation=='NOTREC',])
misclass.survival <- rbind(act.survival[act.survival$recommendation=='NOTREC',],
                           obs.survival[obs.survival$recommendation=='REC',])
```

```
#OBS-REC v OBS-NOTREC with INC
survdiff(Surv(OS, OSCens)~recommendation, data=obs.survival)
plot(survfit(Surv(OS, OSCens)~recommendation, data=obs.survival), col=c(4,6,11),
lt=1:3)
legend(1, 0.3, lty=c(2, 1, 3), col=c(6, 4, 11), legend=c('NOTREC', 'INC', 'REC'))
title(main="OBS-REC v OBS-NOTREC (including INC)")

#OBS-REC v OBS-NOTREC without INC
survdiff(Surv(OS, OSCens)~recommendation,
data=obs.survival[!(obs.survival$recommendation=='CHOICE'),])
plot(survfit(Surv(OS, OSCens)~recommendation,
data=obs.survival[!(obs.survival$recommendation=='CHOICE'),]), col=c(6,11), lt=2:3)
legend(1, 0.3, lty=2:3, col=c(6, 11), legend=c('NOTREC', 'REC'))
title(main="OBS-REC v OBS-NOTREC (without INC)")

#ACT-REC v ACT-NOTREC
survdiff(Surv(OS, OSCens)~recommendation,
data=act.survival[!(act.survival$recommendation=='CHOICE'),])
plot(survfit(Surv(OS, OSCens)~recommendation,
data=act.survival[!(act.survival$recommendation=='CHOICE'),]), col=c(4,6,11), lt=1:3)
legend(1, 0.3, lty=c(2,1), col=c(6,4), legend=c('ACT-REC', 'ACT-NOTREC'))
title(main="ACT-REC v ACT-NOTREC")

#ACT-REC v OBS-REC
survdiff(Surv(OS, OSCens)~TX, data=rec.survival)
plot(survfit(Surv(OS, OSCens)~TX, data=rec.survival), col=c(4,11), lt=1:2)
legend(1, 0.3, lty=1:2, col=c(4, 11), legend=c('ACT-REC', 'OBS-REC'))
title(main="ACT-REC vs OBS-REC")

#ACT-NOTREC v OBS-NOTREC
survdiff(Surv(OS, OSCens)~TX, data=notrec.survival)
plot(survfit(Surv(OS, OSCens)~TX, data=notrec.survival), col=c(4,11), lt=1:2)
legend(1, 0.3, lty=2:1, col=c(11, 4), legend=c('OBS-NOTREC', 'ACT-NOTREC'))
title(main="ACT-NOTREC vs OBS-NOTREC")

# RIGHT v WRONG TX
right.wrong.fit <- survfit(Surv(OS, OSCens)~
                    rep(1:2, c(nrow(correct.survival),
                            nrow(misclass.survival))),
```

```
                    data=rbind(correct.survival, misclass.survival))
plot(right.wrong.fit, col=c(4,11), lt=1:2)
legend(1, 0.3, lty=1:2, col=c(4, 11), legend=c('Right TX', 'Wrong TX'))
title(main="Right TX vs Wrong TX Taken (according to model)")
survdiff(Surv(OS, OSCens)~
         rep(1:2, c(nrow(correct.survival),
                 nrow(misclass.survival))),
       data=rbind(correct.survival, misclass.survival))

right.wrong.5year <- rbind(correct.survival, misclass.survival)
right.wrong.5year[right.wrong.5year$OS > 5,]$OSCens <- 0
right.wrong.5year.survfit <- survfit(Surv(right.wrong.5year$OS,
right.wrong.5year$OSCens) ~ rep(1:2, c(nrow(correct.survival),

nrow(misclass.survival))),
                          data=rbind(correct.survival, misclass.survival))
plot(right.wrong.5year.survfit, col=3:8, lt=1:2)
legend(7.5, 0.3, lty=1:2, col=3:8, legend=c('Correct', 'Misclass'))
survdiff(Surv(right.wrong.5year$OS, right.wrong.5year$OSCens) ~ rep(1:2,
c(nrow(correct.survival),
                                                nrow(misclass.survival))),
       data=rbind(correct.survival, misclass.survival), rho=0)

#########################################################################
#                                          #
#   Subgroup analysis by stage                    #
#                                          #
#########################################################################

#Stage I

correct.survival.stage1 <-
correct.survival[correct.survival$Stage==1|correct.survival$Stage=='1B',]
misclass.survival.stage1 <-
misclass.survival[misclass.survival$Stage==1|misclass.survival$Stage=='1B',]

right.wrong.fit.stage1 <-
  survfit(Surv(OS, OSCens)~ rep(1:2, c(nrow(correct.survival.stage1),
                           nrow(misclass.survival.stage1))),
```

```
        data=rbind(correct.survival.stage1,
                misclass.survival.stage1))
plot(right.wrong.fit.stage1, col=c(4,11), lt=1:2)
legend(1, 0.3, lty=1:2, col=c(4, 11), legend=c('Right TX', 'Wrong TX'))
title(main="Stage 1 Patients: Right TX vs Wrong TX Taken (according to model)")
survdiff(Surv(OS, OSCens)~
         rep(1:2, c(nrow(correct.survival.stage1),
                 nrow(misclass.survival.stage1))),
       data=rbind(correct.survival.stage1, misclass.survival.stage1))

#Stage II

correct.survival.stage2 <-
correct.survival[correct.survival$Stage==1|correct.survival$Stage=='2A'|correct.survival
$Stage=='2B',]
misclass.survival.stage2 <-
misclass.survival[misclass.survival$Stage==2|misclass.survival$Stage=='2A'|correct.surv
ival$Stage=='2B',]

right.wrong.fit.stage2 <-
  survfit(Surv(OS, OSCens)~ rep(1:2, c(nrow(correct.survival.stage2),
                           nrow(misclass.survival.stage2))),
        data=rbind(correct.survival.stage2,
                misclass.survival.stage2))
plot(right.wrong.fit.stage2, col=c(4,11), lt=1:2)
legend(1, 0.3, lty=1:2, col=c(4, 11), legend=c('Right TX', 'Wrong TX'))
title(main="Stage 2 Patients: Right TX vs Wrong TX Taken (according to model)")
survdiff(Surv(OS, OSCens)~
         rep(1:2, c(nrow(correct.survival.stage2),
                 nrow(misclass.survival.stage2))),
       data=rbind(correct.survival.stage2, misclass.survival.stage2))


#########################################################################
#                                                       #
#   Compute measures of model performance                   #
#                                                       #
#########################################################################
```

```
rec.correct <- c()
rec.misclass <- c()
notrec.correct <- c()
notrec.misclass <- c()
act.correct <- c()
act.misclass <- c()
obs.correct <- c()
obs.misclass <- c()
act.choice <- c()
obs.choice <- c()


for(patient in rownames(act.survival)){
  if(act.survival[patient, 'recommendation']=='CHOICE'){
    act.choice <- c(act.choice)
  } else if(act.survival[patient, 'recommendation'] == act.survival[patient, 'given.class']){
    act.correct <- c(act.correct, patient)
    if(act.survival[patient, 'given.class']=='REC'){
      rec.correct <- c(rec.correct, patient)
    } else if(act.survival[patient, 'given.class']=='NOTREC') {
      notrec.correct <- c(notrec.correct, patient)
    }
  } else {
    act.misclass <- c(act.misclass, patient)
    if(act.survival[patient, 'given.class']=='REC'){
      rec.misclass <- c(rec.misclass, patient)
    } else if(act.survival[patient, 'given.class']=='NOTREC'){
      notrec.misclass <- c(notrec.misclass, patient)
    }
  }
}

for(patient in rownames(obs.survival)){
  if(obs.survival[patient, 'recommendation']=='CHOICE'){
    obs.choice <- c(obs.choice)
  } else if(obs.survival[patient, 'recommendation'] == obs.survival[patient, 'given.class']){
    obs.correct <- c(obs.correct, patient)
    if(obs.survival[patient, 'given.class']=='REC'){
      rec.correct <- c(rec.correct, patient)
```

```
    } else if(obs.survival[patient, 'given.class']=='NOTREC') {
      notrec.correct <- c(notrec.correct, patient)
    }
  } else {
    obs.misclass <- c(obs.misclass, patient)
    if(obs.survival[patient, 'given.class']=='REC'){
      rec.misclass <- c(rec.misclass, patient)
    } else if(obs.survival[patient, 'given.class']=='NOTREC'){
      notrec.misclass <- c(notrec.misclass, patient)
    }
  }
}


rc <- length(rec.correct)
nc <- length(notrec.correct)
rm <- length(rec.misclass)
nm <- length(notrec.misclass)
ac <- length(act.correct)
oc <- length(obs.correct)
am <- length(act.misclass)
om <- length(obs.misclass)


#Accuracy
acc <- (rc + nc)/(rc + nc + rm + nm)

#Sensitivity/TPR
tpr <- (rc)/(rc + rm)

#Specificity/TNR
tnr <- (nc)/(nc + nm)

#Precision/PPV: positive predictive value
ppv <- rc/(rc + nm)

#NPV: negative predictive value
npv <- nc/(nc + rm)
```

```
#FPR: False positive rate
fpr <- nm/(nm + nc)

#FNR: False negative rate
fnr <- rm/(rc + rm)

#FDR: False discovery rate
fdr <- nm/(rc + nm)
########################################
#                                      #
#   Apply OBS.RF to test data          #
#    Considering overall               #
#    and 5 year survival               #
#                                      #
########################################
mtry=25
ntree=500

obs.survival.cutoff <- 3.815
test.survival$given.class[test.survival$OS > obs.survival.cutoff] <- 'NOTREC'
test.survival$given.class[test.survival$OS <= obs.survival.cutoff] <- 'REC'

test.obs.rf.data <- data.frame(test.obs.scaled, given.class=test.survival$given.class,
                        OS=test.survival$OS, stringsAsFactors=TRUE)
test.act.rf.data <- data.frame(test.act.scaled, given.class=test.survival$given.class,
                        OS=test.survival$OS, stringsAsFactors=TRUE)

obs.rf <- randomForest(as.formula(paste('given.class ~', paste(colnames(obs),
                                        collapse='+'))), data=obs.rf.data,
                  importance=TRUE, proximity=TRUE, ntree=ntree, mtry=mtry)
act.rf <- randomForest(as.formula(paste('given.class ~', paste(colnames(act.scaled),
collapse='+'))),
                  data=act.rf.data, importance=TRUE, proximity=TRUE, ntree=ntree,
mtry=mtry)

test.obs.predicted <- predict(obs.rf, test.obs.scaled)

test.5year <- test.survival
test.5year[test.5year$OS > 5,]$OSCens <- 0
```

```
test.5year.fit <- survfit(Surv(test.5year$OS, test.5year$OSCens) ~ test.obs.predicted)
plot(test.5year.fit, col=3:8, lt=1:2)
title('OBS Random Forest Classifications: Test Data')
legend(6, 0.3, lty=1:2, col=3:8, legend=c('OBS-good', 'OBS-poor'))
survdiff(Surv(test.5year$OS, test.5year$OSCens) ~ test.obs.predicted, rho=0)

test.act.predicted <- predict(act.rf, test.act.scaled)

reclist <- rep('', 181)
names(reclist) <- rownames(test.survival)
for (patient in rownames(test.survival)){
  if (test.act.predicted[patient] == 'REC' & test.obs.predicted[patient] == 'NOTREC') {
    reclist[patient] <- 'CHOICE'
  } else if (test.act.predicted[patient] == 'NOTREC' & test.obs.predicted[patient] ==
'REC') {
    reclist[patient] <- 'CHOICE'
  } else if (test.act.predicted[patient] == 'REC' & test.obs.predicted[patient] == 'REC') {
    reclist[patient] <- 'REC'
  } else if (test.act.predicted[patient] == 'NOTREC' & test.obs.predicted[patient] ==
'NOTREC') {
    reclist[patient] <- 'NOTREC'
  }
}
test.survival$recommendation <- reclist

test.5year <- test.survival
test.5year[test.5year$OS > 5,]$OSCens <- 0
test.5year.fit <- survfit(Surv(OS, OSCens) ~ recommendation,
data=test.5year[!(test.survival$recommendation=='CHOICE'),])
plot(test.5year.fit, col=3:4, lt=1:2, conf.int=F, xlab="Time (years)", ylab="Survival")
title('Random Forest Model Applied to Test Data, 5-year Survival')
legend(9.5, 0.2, lty=1:2, col=3:8, legend=c('NOTREC', 'REC'))
text(10.5, 0.26, "Log-rank:\n P=0.197, Chi-squared=1.7")
survdiff(Surv(OS, OSCens) ~ recommendation, rho=0,
data=test.5year[!(test.survival$recommendation=='CHOICE'),])


#######################################################
title()
```

```
test.choice <- c()
test.correct <- c()
test.misclass <- c()
rec.correct <- c()
rec.misclass <- c()
notrec.correct <- c()
notrec.misclass <- c()
for(patient in rownames(test.survival)){
  if(test.survival[patient, 'recommendation']=='CHOICE'){
    test.choice <- c(test.choice, patient)
  } else if(test.survival[patient, 'recommendation'] == test.survival[patient, 'given.class']){
    test.correct <- c(test.correct, patient)
    if(test.survival[patient, 'given.class']=='REC'){
      rec.correct <- c(rec.correct, patient)
    } else if(test.survival[patient, 'given.class']=='NOTREC') {
      notrec.correct <- c(notrec.correct, patient)
    }
  } else {
    test.misclass <- c(test.misclass, patient)
    if(test.survival[patient, 'given.class']=='REC'){
      rec.misclass <- c(rec.misclass, patient)
    } else if(test.survival[patient, 'given.class']=='NOTREC'){
      notrec.misclass <- c(notrec.misclass, patient)
    }
  }
}

length(test.correct)
length(test.misclass)
length(test.choice)
length(rec.correct)
length(rec.misclass)
length(notrec.correct)
length(notrec.misclass)
print(length(test.correct)/(181-length(test.choice)))
```

# BIBLIOGRAPHY

BIBLIOGRAPHY

Ahn, Hongshik, and Hojin Moon. 2010. "Classification: Supervised Learning with High-Dimensional Biological Data." In *Statistical Bioinformatics for Biomedical and Life Science Researchers*, edited by Jae K. Lee, 129-156. Hoboken: Wiley-Blackwell.

Bianchi, Fabrizio, et al. 2007. "Survival Prediction of Stage I Lung Adenocarcinomas by Expression of 10 Genes." *Journal of Clinical Investigation*: (117) 3436-3444.

Bioconductor. "R package: affy." Last modified June 19, 2014. Accessed July 5, 2014. http://www.bioconductor.org/packages/release/bioc/html/affy.html.

------. "R package: limma." Last modified October 21, 2013. Accessed August 23, 2014. http://www.bioconductor.org/packages/release/bioc/html/limma.html.

Bou-Hamad, Imad, Denis Larocque, and Hatem Ben-Ameur. 2011. "A Review of Survival Trees." *Statistics Survey*: (5) 44-71.

Breiman, Leo. 2001. "Random Forests." *Machine Learning*: (45) 5-32.

Breiman, Leo, Jerome H Friedman, Richard A Olshen, and Charles J Stone. 1984. *Classification and Regression Trees.* Boca Raton: Chapman & Hall/CRC.

Butts, Charles A., et al. 2009. "Randomized Phase III Trial of Vinorelbine Plus Cisplatin Compared With Observation in Completely Resected Stage IB and II Non-Small-Cell Lung Cancer: Updated Survival Analysis of JBR-10." *Journal of Clinical Oncology*: (28) 29-34.

Chen, Hsuan-Yu, et al. 2007. "A Five-Gene Signature and Clinical Outcome in Non–Small-Cell Lung Cancer." *New England Journal of Medicine*: (356) 11-20.

Der, Sandy D, et al. 2014. "Validation of Histology-Independent Prognostic Gene Signature for Early-Stage, Non-Small-Cell Lung Cancer Including Stage IA Patients." *Journal of Thoracic Oncology*, 2014: (9) 59-64.

Dijk, M R van, E W Steyerberg, S P Stenning, and J D F Habbema. 2004. "Identifying Subgroups Among Poor Prognosis Patients with Nonseminomatous Germ Cell Cancer by Tree Modelling: A Validation Study." *Annals of Oncology*: (15) 1400-1405.

Dziuda, Darius M. 2010. *Data Mining for Genomics and Proteomics.* Hoboken: John Wiley & Sons Inc.

Efron, Bradley, and R.J. Tibshirani. 1993. *An Introduction to the Bootstrap.* Boca Raton: Chapman and Hall/CRC.

Fossa, Sophie D, et al. 2007. "Noncancer Causes of Death in Survivos of Testicular Cancer." *Journal of the National Cancer Institute*: (99) 533-544.

Foster, Jared C, Jeremy M G Taylor, and Stephen J Ruberg. 2011. "Subgroup identification from randomized clinical trial data." *Statistics in Medicine*: (30) 2867-2880.

Grunkemeier, Gary L, and Ruyun Jin. 2001. "Receiver Operating Characteristic Curve Analysis of Clinical Risk Models." *Annals of Thoracic Surgery*: (72) 323-326.

Hess, Kenneth R, Marie C Abbruzzese, Renato Lenzi, Martin N Raber, and James L Abbruzzese. 1999. "Classification and Regression Tree Analysis of 1000 Consecutive Patients with Unknown Primary Carcinoma." *Clinical Cancer Research*: (5) 3403-3410.

Kollmannsberger, C, C Nichols, C Meisner, F Mayer, L Kanz, and C Bokemeyer. 2000. "Identification of prognostic subgroups among patients with metastatic 'IGCCCG poor-prognosis' germ-cell cancer: An explorative analysis using cart modeling." *Annals of Oncology*: (11) 1115-1120.

Leong, David, Rajat Rai, Brandon Nguyen, Andrew Lee, and Desmond Yip. 2014. "Advances in adjuvant systemic therapy for non-small-cell lung cancer." *World Journal of Clinical Oncology*: (5) 633-645.

Loh, Wei-Yin. 2011. "Classification and regression trees." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*: (1) 14-23.

Lu, Yan, et al. 2006. "A Gene Expression Signature Predicts Survival of Patients with Stage I Non-Small Cell Lung Cancer." *PLoS Medicine*: (3) 2229-2243.

Luo, J, et al. 2010. "A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data." *Pharmacogenomics Journal*: (10) 278-291.

National Cancer Institute. "Non-Small Cell Lung Cancer Treatment." Last modified 2015. Accessed February 1, 2015. http://www.cancer.gov/cancertopics/pdq/treatment/non-small-cell-lung/healthprofessional/page7.

National Center for Biotechnology Information. "About GEO2R." Last modified 2014. Accessed August 23, 2014. http://www.ncbi.nlm.nih.gov/geo/info/geo2r.html.

------. "Gene Expression Omnibus Series GSE14814." Last modified February 12, 2009. Accessed May 16, 2014. http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=gse14814.

------. "Gene Expression Omnibus Series GSE50081." Last modified August 21, 2013. Accessed August 18, 2011. http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE50081.

------. "GEO2R: GEO Series Interactive Analyzer." Last modified 2014. Accessed August 23, 2014. http://www.ncbi.nlm.nih.gov/geo/geo2r/.

Raponi, Mitch, et al. 2006. "Gene Expression Signatures for Predicting Prognosis of Squamous Cell and Adenocarcinomas of the Lung." *Cancer Research*: (66) 7466-7472.

Steck, Harald, and Tommi S. Jaakkola. 2003. "Bias-corrected Bootstrap and Model Uncertainty." In *Advances in Neural Information Processing Systems*, 521-528. Cambridge: MIT Press.

Tibshirani, Robert, Jerome Friedman, and Trevor Hastie. 2009. *Elements of Statistical Learning.* New York City: Springer.

Tsai, Chen-An, Dung-Tsa Chen, James J Chen, Charles M Balch, John F Thompson, and Seng-Jaw Soong. 2007. "An Integrated Tree-Based Classification Approach to Prognostic Grouping with Application to Localized Melanoma Patients." *Journal of Biopharmaceutical Statistics*: (17) 445-460.

Winton, Timothy, et al. 2005. "Vinorelbine plus Cisplatin vs. Observation in Resected Non-Small-Cell Lung Cancer." *New England Journal of Medicine*: (352) 2589-2597.

Zhu, Chang-Qi, Keyue Ding, Dan Strumpf, and Matthew Meyerson, Nathan Pennell, Roman K. Thomas, Katsuhiko Naoki, Christine Ladd-Acosta, Ni Liu, Melania Pintilie, Sandy Der, Lesley Seymour, Igor Jurisica, Frances A. Shepherd, and Ming-Sound Tsao Barbara A. Weir. 2010. "Prognostic and Predictive Gene Signature for Adjuvant Chemotherapy in Resected Non-Small-Cell Lung Cancer." *Journal of Clinical Oncology*: (28) 4417-4424.