USING DNA METHYLATION TO PREDICT WHITE BLOOD CELLS FREQUENCIES IN TUMOR TISSUE SAMPLES

CS 229 – FINAL PROJECT WRITE-UP – FALL 2015

Marcos M. Prunello
Biomedical Informatics MS Student
marcosp@stanford.edu

Ying-Cheng Chris Lee
Electrical Engineering MS Student
leechris@stanford.edu

Abstract

Large multi-cancer data sets are available that profile the epigenome for tumor tissue samples. However, such cancer samples are made up of a mixture of cell types and it is necessary to determine to which extent the samples are composed by tumor cells or by other cell types, such as blood cells. We built models that are able to estimate the frequency of different types of blood cells present in brain tumor tissue samples using only DNA methylation data registered on the samples, and marked the most important genes for this task.

INTRODUCTION

Cancer refers to a group of diseases characterized by an uncontrolled growth in cells which can extend to different parts of the body, generally caused by changes in DNA. There are about 14.1 million new cases per year in the world and it causes about 14.6% of all human deaths. In the United States, the financial costs of cancer have been estimated at $1.16 trillion dollars per year. There are many different types of cancers, each with its own subtypes.

The importance of epigenomics when studying cancer cannot be understated. Epigenomics is the study of epigenetic modifications, which include all the variations that affect gene expression without altering the DNA sequence. Changes on top of DNA have shown to activate oncogenes or deactivate tumor suppressor genes. Large multi-cancer data sets are now available that profile the epigenome for tumor tissue samples. Nevertheless, such cancer samples are made up of a mixture of cell types. Mathematical approaches are needed to deconvolve these mixtures and determine to which extent the samples are composed by tumor cells or by other cell types, such as blood cells.

DNA methylation, an epigenetic process consisting of a chemical modification of DNA, is responsible for cellular differentiation and hence can be used to distinguish distinct cell lineages.

Our goal in this project was to predict the proportion of different blood cell types in tumor tissue samples (our continuous response) using DNA methylation data measured for an extensive set of genes (our predictors, also continuous), and to identify which are the genes (the features) that are most important for this prediction task. To achieve this, we tried several machine learning methods: linear regression with penalization, K-nearest neighbors, support vector regression and regression trees (basic trees, but also random forests, bagging and boosting).

RELATED WORK

The use of methylation signatures to estimate the distribution of blood cell types in a sample is called deconvolution of DNA methylation and it involves learning the specific methylation signature of each cell type from a set of samples of purified known cell subtypes representing gold-standard data, to later infer the type of cells present in the target samples. Some research in this topic has been carried out lately, but it stills remains a non-trivial problem with no clear solution. Houseman [1] described a method which resembles regression calibration, where a methylation signature is considered to be a high dimensional multivariate surrogate for the distribution of blood cells. Jaffe [2] tailored Houseman's method using a new reference set produced with 450k microarrays instead of 27k that was published by Reinius [3].

In a first stage of our work we applied Houseman's method in the same way as Jaffe. Since this method was developed for DNA methylation deconvolution in blood samples but we applied it in tissue samples, we looked for some way of validation of these results. We decided to perform gene expression deconvolution on the same samples with a well established and validated method, Cibersort [4]. We found no match between our results from the deconvolution of DNA methylation and those from Cibersort's gene expression deconvolution. Facing these negative results, we decided to approach the problem as described in the introduction, using the DNA methylation data to predict the frequencies of the different blood cell types, using as ground truth the

frequencies provided by Cibersort through its gene expression deconvolution.

DATA SET AND FEATURES

Gliobastoma cancer data

We decided to focus our analysis in Glioblastoma (GBM), which is the most common and most aggressive malignant primary brain tumor. We extracted methylation data for GBM cancer samples from The Cancer Genome Atlas (TCGA). Illumina Infinium HumanMethylation 27k were used to produce TCGA's DNA methylation data, which is quantified with beta values in a range from 0 to 1 representing the proportion of methylation signal versus total signal. Values close to 1 represent high levels of DNA methylation and values close to 0 low levels. Those CpG sites with more than 10% missing values in all samples were removed and 15-K Nearest Neighbor (KNN) algorithm was applied to estimate the rest of missing values [5]. TCGA samples were analyzed in batches so we used Combat to adjust for this effect [6]. We also extracted matched gene expression data, produced with microarrays technology. Data was log-transformed, infinities were replaced with a low value and missing value estimation and batch correction were applied as described before. The final data set consisted of 321 samples.

Response features

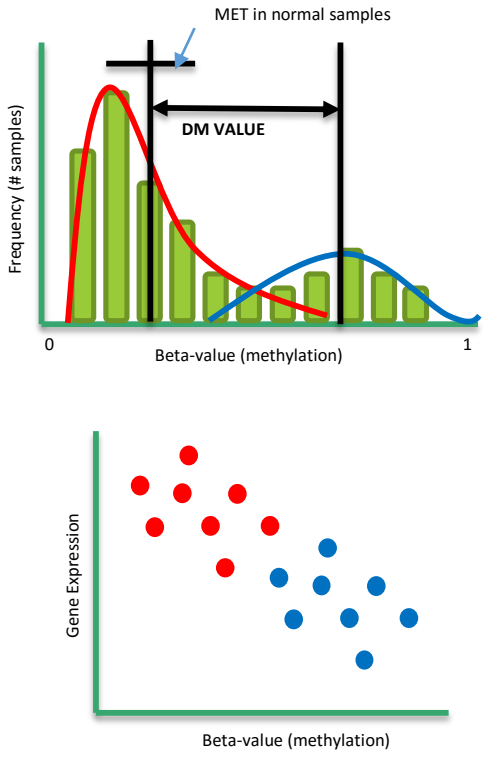
In our prediction task, the response is the proportion of seven different blood cell types in the tumor tissue samples: D4+ T cells, CD8+ T cells, CD56+ NK cells, CD19+ B cells, CD14+ monocytes, neutrophils, and eosinophils. We generated our response features using Cibersort [4], an algorithm for gene expression deconvolution which accurately quantifies the relative levels of distinct cell types within a complex gene expression admixture. Such mixtures can derive from malignant or normal solid tissues.

Predictors

Our predictors are the methylation measurements (beta values) on a total of 17812 genes for each of the 321 samples. With this data set, we faced two challenges. First, methylation data is known to be noisy, and second, we needed to reduce the number of available features (genes) though doing some sort of variable selection. We decided to approach these two issues analyzing the set of predictors with an algorithm called MethylMix [7], which identifies hypo- and hyper-methylated genes in cancer samples that are also predictive of transcription through the following three steps (Figure 1).

First, for each gene a beta mixture model is fitted to the methylation beta values to generate subgroups of samples with similar DNA methylation levels, using an Expectation Maximization algorithm for mixtures of beta distributions (since beta-values represent the proportion of methylation). Starting with a model with only one component, a new beta mixture component is added iteratively if it improves the fit of the model according to the Bayesian Information Criterion (BIC) for model selection. Each of the final mixture components represents a subset of patients with a common methylation profile (a "methylation state"). Secondly, for each of these subgroups the difference between its mean methylation and the mean methylation in normal tissue samples is calculated and it is called DM-value (Differential Methylation). Only those genes with significant DM-values are selected for the third step, in which a linear regression is used to model the expression of each gene by its own methylation. If there is a significant negative association between methylation and gene expression, the gene is finally reported as a methylation-driven genes.

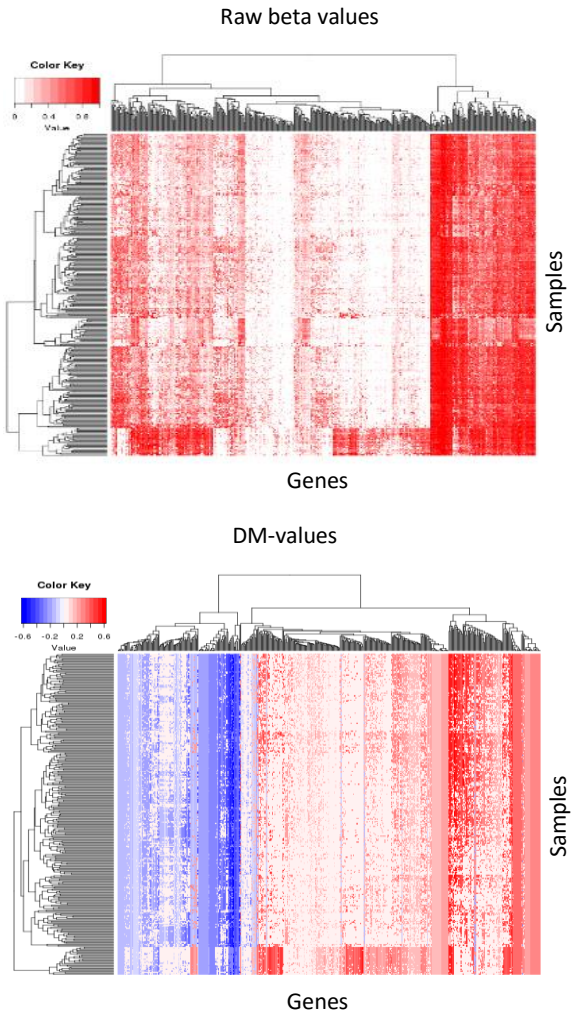
Figure 1. Representation of MethylMix algorithm. This is a methylation-driven gene because it shows methylation states that differ from variation in normal samples and also its methylation profile has an effect in gene expression. Each sample receives a DM value, which is the difference between mean methylation in its group and mean methylation in normal samples.



Applying this algorithm to the GBM data, from the original set of 17812 genes we identified 455 methylation-driven genes which constituted our reduced set of predictors.

Furthermore, each sample for each gene can be represented by the DM-value of the group to which belongs, in addition to its original raw methylation value (beta value). DM-values were shown to perform better than raw methylation in some scenarios, provided that they constitute a less noisy version of the data [8]. In this way, we had two matrices of predictors, one with the original methylation values and other with the DM-values, for each of the 455 MethylMix genes (Figure 2). We trained our predictive models in both.

Figure 2. Heatmaps of the 321 samples and 455 genes for both raw data and DM-values



METHODS

We divided our 321 samples into one training set (80%) and one test set (20%). We trained several machine learning methods to predict the frequencies of each of the 7 blood cell subtypes individually, using both the raw methylation and the DM-values training sets. We tuned the parameters required for each method with 10-fold cross validation (CV), and once selected the optimal parameters, we refitted the model in the whole training set. Finally, we predicted the response in the test set, and evaluated the performance with Mean Squared Error (MSE). All the analysis were performed in R [9]. In the following paragraphs we described the methods that we used. In

K-Nearest Neighbors Regression (KNNR)

Given a value for K and a prediction point, KNNR identifies the K training observations that are closest to the given point and predicts its response with the average of the response of the K closest data points. This is a non-parametric method. The optimal number K was selected with 10 fold CV. This method was implemented with the R package "caret" [10].

Linear regression with L1 norm penalization

This method, also known as LASSO, fits a penalized linear regression model, where the coefficients are penalized using L1 norm. This forces some of the coefficient estimates to be zero when the tuning parameter λ is large enough. Hence, this model is in itself a variable selection method, yielding a sparse model, very convenient in our application provided that we have 455 predictors and 321 samples. The objective function being minimized is:

$$\sum_{i=1}^{m}\left(y_i-\beta_0-\sum_{j=1}^{n}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{n}|\beta_j| \qquad (1)$$

We selected the penalization parameter λ with 10 fold CV. This method was implemented using the R package "glmnet" [11].

Support Vector Regression (SVR)

SVR is an extension of Support Vector Machines (SVM) for cases where the response is continuous instead of binary. SVR contains all the main features that characterize SVM, with a loss function that ignores errors situated within a certain distance of the true value. This type of function is often called epsilon intensive loss function, since a margin of tolerance ϵ is set. The dual representation of the model is given by:

$$\min_{\alpha}\frac{1}{2}(\alpha-\alpha^*)^T Q(\alpha-\alpha^*)+\epsilon\sum_{i=1}^{m}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{m}y_i(\alpha_i-\alpha_i^*) \qquad (2)$$
$$\text{such that: } 0\le\alpha_i,\alpha_i^*\le C, i=1,\dots,m, \textstyle\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0$$

where C is the upper bound for the cost parameter, Q is an mxm positive semidefinite matrix, $Q_{ij}=y_i y_j K(x_i,x_j)$, $K(x_i,x_j)=\phi(x_i)^T\phi(x_j)$ is a kernel and m is the number of observations. We implemented SVR with radial or polynomial kernel, but also with no kernel at all (Table 1). In each case, the required parameters were selected with 10 fold CV. SVR was implemented with the R package "e1071" [12].

Table 1. Kernels applied in SVR

Kernel	Formula	Parameters tuned with CV
Linear	$u^T v$	C, ϵ
Radial	$\exp(-\gamma \|\|u - v\|\|^2)$	C, ϵ, γ
Polynomial	$\gamma(u^T v)^d$	C, ϵ, γ, d

Tree-based methods

Tree-based methods for regression involve segmenting the predictor space into a number of simple regions and using the mean of the training observations in the region to which a point that we want to predict belongs. We first fitted a basic regression tree and pruned it with 10 fold CV. Since basic regression trees do not always provide the best results in terms of accuracy, we also applied bagging, random forests, and boosting to improve the performance. Each of these approaches involves producing multiple trees which are then combined to yield a single prediction. Bagging takes B bootstrap samples of the training observations, fits several regression trees then averages the predicted values, helping to reduce the characteristic high variance in a basic regression tree. Random trees contributes to decorrelate the trees built by bagging, by taking a random sample of some predictors as split candidates each time a split in a tree is considered. Boosting is another which in the context of regression trees involves growing the trees sequentially, each tree is grown using information from previously grown trees. The number of trees as well as other parameters like the number of predictors chosen randomly in random forests were optimized looking at out-of-bag (OOB) error estimates. We used the R packages "tree" [13], "randomForests" [14] and "gbm" [15].

RESULTS

A total of 126 models were trained, given our 7 cell subtypes, 2 data types, and 9 candidate models (KNNR, Lasso, SVR with 3 different kernels and 4 tree-based methods). After optimizing the parameters for each model with 10 fold CV, the model was re fit in the whole training set and predictions were obtained for both the training and the independent test set. Then we calculated different measurements to evaluate the performance of the model but we used the traditional Mean Square Error (MSE) on the test set to select the best performing model for each subtype (Table 2).

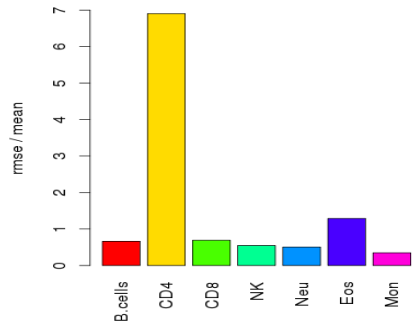
Looking not only at the best performing model but also to the top models for each cell subtype, we did not find that either the raw methylation data or the DM-values outperformed the other data type in most cases. Although we hypothesized that the DM-values could performed better as they represent a de-noised version of the data, it seems that the raw methylation data was enough for the prediction task.

Table 2. Top performing model for each cell subtype

Subtype	Obs Mean	Best Model	MSE Test	MSE Train	Cor Test	Cor Train
B cells	0.0941	Raw - SVR Lin	0.0043	0.002	0.545	0.7915
CD4	0.0045	Raw - Lasso	0.0007	0.0003	0.3907	0.8812
CD8	0.1733	DM -Rand Forests	0.0181	0.0032	0.3948	0.9877
Eos	0.1404	Raw - Boosting	0.0045	0.0039	0.1286	0.559
Mon	0.2449	DM - SVR Linear	0.0095	0.0089	0.2716	0.6221
Neu	0.0547	Raw - SVR Radial	0.0143	0.0076	0.4331	0.8581
NK	0.2882	Raw - SVR Polyn	0.0053	0.0038	0.6236	0.7769

In terms of the models, different methods worked better for different cell subtypes, but in 4 out of the 7 subtypes it was support vector regression the one that achieved the best results. For B cells and Monocytes, the best model did not require any kernel, whereas for Neutrophils and Natural Killers used a radial or a polynomial kernel, respectively. For CD8 and Eosinophils, random forests and boosting performed the best. They represent improvements over the basic regression tree, since both fit several trees. Random forests takes random samples of the predictors to make a final prediction using not very correlated trees, and boosting in each new tree weights more those points with bigger errors in the previous trees, so it is reasonable to see that these methods worked better than the basic regression tree. For CD4, the best performing model is LASSO. Nevertheless, results for CD4 should be interpreted with care, since the frequency of CD4 in the samples is very low and the response vector consists mainly of zeros and some small frequencies. In fact, calculating the ratio of the squared root of MSE (RMSE) and the mean of the true observed frequency, it is possible to see that among all subtypes, CD4 is the one with poorest performance (Figure 3). KNNR did not appear in any case to be among the top performing models. Although this simple algorithm works well in some contexts, it usually performs worse in very high dimensional spaces with few samples like in our problem.

Figure 3. Root Mean Square Error relative to mean value of the response in the test set.



We also calculated the correlation between the predicted values and the observed values for both the training and the test set. The correlations in the training set are strong, and some models not finally reported here presented even higher correlations. However, in the test set they are

weaker, showing that there is room for improvement in the predictions. Even though we selected our reduced feature set independently of the response, there might be some overfitting in our models, considering as well that the number of predictors is still high in relation to the number of samples.

We identified for each subtype which were the most important genes in the top performing model and we report here the top 5. Again, results for CD4 should be interpreted with care, since the performance was weak, and also because the L1 penalty of the LASSO forces some coefficients to be zero, and if there are some correlated genes, arbitrarily one is kept as a non-zero coefficient and the others become zero. We intersected the top 30 genes for each subtype and found that there were no genes pointed as important in most subtypes simultaneously.

Table 3. Top performing model for each cell subtype.

Subtype	Top 5 important genes				
B cells	HSD17B14	ZNF329	PLA2R1	MT2A	PDGFRA
CD4	ABCA5	ABHD8	ACAA2	ACN9	ACOT4
CD8	TNFRSF1A	KLHL26	PSMB8	CTSZ	OAS1
Eos	CRYBB1	DDX43	C17orf76	ACSBG1	ANKRD9
Mon	KLHL34	SMOC1	RDH5	CABYR	ZNF135
Neu	NETO1	STAG3	ME1	FOXD1	CTHRC1
NK	ZNF549	BCL11A	C9orf95	ZNF781	DUSP23

The resultant top frequency predictive genes suggests some primitive principles. The fact that very few top gene overlaps between different white blood cell types suggests very differentiated activities between the cell types. It is possible to see that each cell types' top genes seem to showcase execute main and various biochemical tasks known to be specific to that particular cell type, for example, CRYBB1 beta-crystallin B1 for Eosinophils known having crystalloid core, and others. On the context of gene regulation, each cell types contain one top gene that is of DNA-binding functionality: ZNF329 for B cells, ABCA5 for CD4+ T cells, OAS1 for CD8+ T cells, DDX43 for Eosinophils, ZNF135 for Monocytes, ME1, STAG3 and FOXD1 for Neutrophils, ZNF549 and ZNF781 for Natural Killer cell. These might be of interest for implications on gene regulations. On the context of possible tumor immunological functions or apoptotic function, a few genes stand out: TNFRSF1A, tumor necrosis factor superfamily 1A for CD8+ T cells, ANKRD9, Ankyrin repeat domain-containing protein 9 for Eosinophils, STAG3, cohesin subunit SA-3 for Neutrophils, BCL11A, B cell CLL/lymphoma 11A isoform2 for Natural Killer cells. These may be of interest for implications on cancer immnological functions. Finally a few top genes appear to be less annotated with Gene Ontology terms: C17orf76 for Eosinophils, KLH34 for Monocytes, C9orf95 and ZNF781 for Natural Killer cells. These might be genes with novel implications.

DISCUSSION AND CONCLUSION

In this project we built models that are able to predict the frequency of 7 blood cell subtypes (D4+ T cells, CD8+ T cells, CD56+ NK cells, CD19+ B cells, CD14+ monocytes, neutrophils, and eosinophils) in samples from brain tumor tissue using only DNA methylation data. We applied an algorithm which identifies methylation driven genes to select a reduced feature space, and trained our models with both raw methylation data and DM-values. Although we suspected that the DM-values would show a better performance as they constitute de-noised data, in 5 out of the 7 subtypes, the best model used the raw data.

In relation to the performance of the models, support vector regression performed was selected as the best model for 4 subtypes, although with different kernels. Probably the ability of SVR to work in high dimensional spaces contributed to this outstanding performance. In two cases the best model was a tree-based method and in one the penalized linear regression. KNN was not picked as one of the best models, this simple method is known not to perform very well in highly dimensional spaces, and hence our higher number of predictors than samples may not be most adequate context for KNN.

From the inspection of the results, it is clear that there is wide opportunity for improvement of the performance of this prediction task. For example, our feature selection with MethylMix was carried out independently from the response; a future modification should explore in more detail the set of 17812 genes with some variable selection procedure that also involves the response. Furthermore, we did not account for the fact that our response was a proportion, ranging from 0 to 1, and the addition of this constraint may strengthen our models, for example, the linear regression model which as it is can predict values outside of the plausible range.

Other aspect of this problem that should be addressed is the joint prediction of our 7 responses, and not individually as we did. Since the relative frequencies of each cell subtype are related to each other, it seems reasonable to predict all the frequencies together with a multivariate response. Also, we acknowledge that our data set was rather small, 321 samples divided into the training and test sets. We would like to evaluate the results of our work in larger data sets, from other types of tumor as well.

Even though this work can be further extended with the suggestions previously mentioned, we were able to see that DNA methylation provides clear signal of the immune composition of the tumor tissue and can be used to infer the relative frequencies of different types of white blood cells present in the samples.

REFERENCES

[1] E. A. Houseman, W. P. Accomando, D. C. Koestler, B. C. Christensen, C. J. Marsit, H. H. Nelson, J. K. Wiencke, and K. T. Kelsey, "DNA methylation arrays as surrogate measures of cell mixture distribution.," *BMC Bioinformatics*, vol. 13, no. 1, p. 86, 2012.

[2] A. E. Jaffe and R. a Irizarry, "Accounting for cellular heterogeneity is critical in epigenome-wide association studies.," *Genome Biol.*, vol. 15, no. 2, p. R31, 2014.

[3] L. E. Reinius, N. Acevedo, M. Joerink, G. Pershagen, S.-E. Dahlén, D. Greco, C. Söderhäll, A. Scheynius, and J. Kere, "Differential DNA methylation in purified human blood cells: implications for cell lineage and studies on disease susceptibility.," *PLoS One*, vol. 7, no. 7, p. e41361, 2012.

[4] A. M. Newman, C. L. Liu, M. R. Green, A. J. Gentles, W. Feng, Y. Xu, C. D. Hoang, M. Diehn, and A. a Alizadeh, "Robust enumeration of cell subsets from tissue expression profiles.," *Nat. Methods*, no. MAY 2014, pp. 1–10, 2015.

[5] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae)," *Proc. Natl. Acad. Sci.* , vol. 100 , no. 14 , pp. 8348–8353, 2003.

[6] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical Bayes methods.," *Biostatistics*, vol. 8, no. 1, pp. 118–27, 2007.

[7] O. Gevaert, "MethylMix: an R package for identifying DNA methylation-driven genes.," *Bioinformatics*, vol. 31, no. 11, pp. 1839–41, 2015.

[8] O. Gevaert, R. Tibshirani, and S. K. Plevritis, "Pancancer analysis of DNA methylation-driven genes using MethylMix," *Genome Biol.*, vol. 16, no. 1, p. 17, 2015.

[9] R Core Team, "R: A Language and Environment for Statistical Computing," *R Found. Stat. Comput. Vienna, Austria.*, 2015.

[10] M. Kuhn, "caret Package," *J. Stat. Softw.*, vol. 28, no. 5, pp. 1–26, 2008.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent.," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[12] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, "Misc functions of the Department of Statistics (e1071), TU Wien," *R package version 1.6-2*, 2014. [Online]. Available: http://cran.r-project.org/package=e1071.

[13] B. Ripley, "tree: Classification and Regression Trees. R package version 1.0-36," 2015. .

[14] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[15] G. Ridgeway, "gbm: Generalized Boosted Regression Models," 2015. [Online]. Available: http://cran.r-project.org/package=gbm.