Boosted machine learning: A simulation study in disease phenotype prediction with single nucleotide polymorphism gene expression

Quinton Neville, MS[1]

[1] *Columbia University Mailman School of Public Health, Department of Biostatistics, 722 West 168th St. NY, NY 10032*

Department of Biostatistics
Mailman School of Public Health
Columbia University
New York, New York

May 4, 2020

1 Introduction

Disease prediction in Genome Wide Association Studies (GWAS) is plagued by Bellman's infamous curse of dimensionality [2, 19]. The high dimensional predictor space that often accompanies genomic expression induces a host of computational and analytic issues when investigating relationships with disease phenotypes (observable characteristics). These micro-array data, collected via mass spectrometry and high-throughput screening, are processed and distilled to the continuous log-scale expression of 100,000's of functional units called Single Nucleotide Polymorphisms (SNP's) [17]. From this massive array of information, investigators have utilized a wide variety of statistical and machine learning tools to characterize the predictive ability of SNP's on disease phenotype outcomes. In this study, we specifically delve into the world of *boosted machine learning algorithms*; conducting a simulation study to analyze the predictive capability of these algorithms with *synthetic SNP expression-phenotype data* and applying these models to a real-world example involving *Down syndrome in mice*.

1.1 Boosting

Given the high dimensional predictor space associated with SNP expression, penalized regression and tree-based algorithms have historically been a focal point of study in phenotype classification [11]. However, modern developments in ensemble machine learning, methods which employ a collection of "weak" learning algorithms like generalized linear models or decision trees, have dramatically increased investigators' ability to predict disease phenotype by SNP expression [11, 21, 22, 31]. In this study, however, we restrict our focus to the class of *boosting algorithms* – which in the context of SNP-disease prediction, "have features of robustness against model misspecification, and... resistance to model overfitting" [11]. Specifically, we investigate 6 primary algorithms:

1. Lasso Regularized Generalized Linear Models (glmnet) for baseline reference
2. Adaptive Boosting (ADA) with decision trees
3. Gradient Boosting Machines (GBM), generalized boosting with decision trees
4. * Extreme Gradient Boosting Machines (xgbTree) with trees
5. * Extreme Gradient Boosting Machines (xgbLinear) with regularized glm's
6. ** Deep Boosting (deepboost)

* *Newer methods with potentially significant application in genomics*

** *New method not well characterized in genomic context*

Structurally, the lasso algorithm is utilized as a baseline learning algorithm [18], while ADA and GBM embody foundational boosting algorithms for investigation [9, 26]. We then compare these well-established methods with the much more recent and complex Extreme GBM (xgBoost) family of algorithms [6]. Lastly, we elect to explore a novel Deep Boosting algorithm developed at Google, which has not been well characterized in the context of genomics and could provide an improved diagnostic tool compared to previous methods [8].

1.1.1 Theoretical Comparison

Boosting, which can be thought of as an adaptive extension of bagging [4], is an ensemble approach which takes a hypothesis class of weak learners, fits the weak learner to a resampling of the original data, and repeats this process until an optimal number of rounds has been reached to minimize empirical error [24, 28]. Unlike bagging however, which employs a uniform bootstrap resampling procedure at every iteration, boosting is adaptive in that it weights learners by their previous mistakes and can resample the data dependent on how many mistakes previous classifiers have made on those examples.

Starting with Adaptive Boosting, from the initial discrete AdaBoost developed by Friedman et. al. in 1996 [12], we employ the gentle, stochastic gradient boosting with decision trees which can be viewed as a "hybrid bagging and boosting algorithm" [14]. Additionally, we also investigate the more generalized form of adaptive boosting coined Generalized Boosting Models or Gradient Boosting Machines (GBM), of which the aforementioned AdaBoost is a special case [24, 26]. These algorithms have been previously implemented and studied in the context of genomic disease prediction, with promising results [11, 21, 22, 31].

Extreme Gradient Boosting (xgBoost) is a more recent development in the world of machine learning, building off of the work of ADA and GBM boosting. On the back end, the methodology and target are identical to previous boosting algorithms, but with additional computational features and complexity. With respect to boosting decision trees specifically, the most notable xgBoost computational improvements are the implementation of an *exact greedy algorithm* to find the optimal splitting plane, a novel approximation method to that greedy algorithm, 'sparsity-aware' split finding, a column block design for parallel learning, and cache-aware access [5, 6]. Previous studies suggest that while these algorithms are significantly more sophisticated than classical AdaBoost or GBM algorithms, they may not actually perform better in the context of SNP-disease phenotype learning [21].

The final, and most complex, model under investigation is the Deep Boosting (deepboost) algorithm developed at Google [8]. Unlike previous learning models, which are restricted to a particular, relatively small hypothesis class, deepboost incorporates a convex ensemble of

multiple, complex hypothesis classes. The derivation of the data-dependent uniform error bound, in the face of increasing Rademacher complexity and VC dimension "is quite remarkable", with full statement and proof found in [8]. Compared to AdaBoost, GBM, or xgBoost, Deep Boosting allows for the consideration of much richer, more complex hypothesis classes – with theoretically better performance on the margins and a potentially reduced risk for over fitting.

2 Methods

This study consists of two main components: (1) a simulation study evaluating the performance of the learning algorithms on synthetic SNP-phenotype data, and (2) a real-world evaluation of their performance from a GWAS regarding Down syndrome in mice. Each algorithm is first tuned to the training data, the optimal parameters are stored, then model diagnostics are computed with repeated 5-fold cross validation (CV), with overall performance being evaluated by Area Under the Receiver Operator Curve (AUC, ROC) and misclassification/error. For the Down syndrome example, data are first separated into an 80/20 training-test split, diagnostics performed on the training set, with one single final evaluation on the 'never-before-seen' testing data. All statistical programming, analysis, and visualizations are performed in *R: A Language and Environment for Statistical Computing* [7, 10, 13, 15, 16, 20, 23, 25, 32].

2.1 Synthetic SNP-Disease Data Generation

Simulating a genetic experiment is an incredibly non-trivial task, and there is an entire body of work devoted entirely to synthetically generating mass spectrometry, high throughput experimental results. Consequently, in this study we restrict our focus to synthetically generating the final result of an '-omics' experiment – the SNP expression of particular families of genes. Previous studies note that "gene expression between cells has been described as being log-Normal or Gamma distributed... [and] an advantage of the Gamma model is that its parameters relate directly to gene burst frequency and size" [3,30,33]. Indeed, many GWAS employ the gamma distribution for specifying SNP models, and for this study, we assume that SNP expression follows a gamma distribution [1, 27, 29].

Next, in addition to the curse of dimensionality, differential SNP gene expression is often highly clustered and/or correlated. This study employs the *simstudy* R package [15] to first generate a pre-specified number of cluster-correlated gamma-distributed SNP expression markers, with randomized mean expression level $\mu \in (0.1, 1)$ and precision $\tau = \sigma^{-2} \in (1, 2)$,

which are then fitted to a synthetic binary disease outcome via a probabilistic logit link. Finally, to investigate the performance of the aforementioned learning algorithms in the context of genomics, we focus on varying the inter-cluster marker correlation $\rho \in \mathbb{R}^{(-1,1)}$, sample size $N \in \mathbb{N}$, number of SNP markers $q \in \mathbb{N}$, and method for linking the SNP gene-expressions to a disease outcome. Thus, in an attempt to replicate a GWAS experiment, we generate 180 case-control data sets varying

a. Sample size – $N = \{100, 250, 500\}$
b. Number of SNP markers – $\{q = 25, 50, 100\}$
c. Correlation – $\rho = \{0.1, 0.3, 0.5, 0.7, 0.9\}$
d. Disease-link method
    i. Linear – $\log\left(\frac{p}{1-p}\right) = \boldsymbol{X}\boldsymbol{\beta}$
    ii. Quadratic – $\log\left(\frac{p}{1-p}\right) = \boldsymbol{X}\boldsymbol{\beta}_1 + \boldsymbol{X}^2\boldsymbol{\beta}_2$
    iii. Sinusoidal – $\log\left(\frac{p}{1-p}\right) = \sin(\boldsymbol{X})\boldsymbol{\beta}_1 + \cos(\boldsymbol{X})\boldsymbol{\beta}_2$
    iv. Power – $\log\left(\frac{p}{1-p}\right) = \sum_{i=1}^{q} \beta_{1i} X_i^{\beta_{2i}}$

Here, we constrain $p = P(Disease) \in (0.4, 0.6)$ to mimic a generally balanced case-control study and constrain ρ to be positive as negative correlation between SNP gene expression is rarely observed. Synthetically, we link the markers to the binary disease with both linear and non-linear functions to investigate the difference in model performance by varying true relationships. Lastly, we fix the number of "true" predictors in the disease-link model at 5, randomly selected from the entire predictor set (which includes the possibility that two "true" markers are in the same correlated-cluster, which holds for complex diseases in practice), and fix the cluster sizes to be 20% of the sample size N.

2.2 Down Syndrome in Mice

Following the simulation, we investigate the predictive capability of these boosting algorithms on a real-world example from a case-control GWAS in mice. Here, the disease outcome of interest is Down syndrome, with roughly half of the observed mice eliciting the disease phenotype. After data cleaning and handling of missing data, we conduct the final case analysis on 1077 mice, each with 68 observed SNP gene-expression features which produce detectable signals in the nuclear fraction of cortex related to neural functioning.

3 Results

3.1 Simulation

First, the synthetic data was observed to represent the 'true' mice data reasonably well, as visualized in Appendix A.1 Figure 4, with 25 synthetic SNP's ($N = 100, \rho = 0.3$) versus the 68 real SNP's in 1077 mice. After tuning all six LASSO, ADA, GBM, xgbTree, xgbLinear, and deepboost algorithms, to all 180 synthetic data sets of varying parameters, the resulting cross validated error and AUC results are displayed in Appendix A.1 Figures 5-7 (error) and Figures 8-10 (AUC).

3.1.1 Error

Considering A.1 Figures 4-6 as a whole, the 5 boosting models performed quite similarly, while LASSO was observed to perform very well for linear and power disease-links, but poorly for quadratic and sinusoidal links. Additionally, in most cases, all boosting models were relatively robust to increasing SNP-cluster correlation, however error rates fluctuated consistently around $\rho = 0.5$ and saw larger amplitudes of error fluctuation for sinusoidal and linear disease-links than quadratic or power. In general, error rates increased both as the number of SNP's grew and correlation magnitude increased. However, in certain cases, we observed a bell shape with lower error rates at low and high correlation versus moderate. Finally, though error rates decreased as sample size increased, as expected, the same fluctuations by correlation and number of SNP's dampened but were persistent with increasing sample size.

Restricting scope specifically to the boosting algorithms of interest, over all simulations ADA and GBM consistently outperformed xgbTree, xgbLinear, and deepboost on average. As sample size grew, we observed that these boosting algorithms were indeed robust to increasing number of SNP's, implying that they may provide a reliable method for SNP-disease prediction resistent to overfitting. Lastly, we actually observed that deepboost performed the worst of the boosting algorithms on average, but with slightly higher precision (lower variance). Initial simulation CV error results imply that, overall, ADA and GBM may be preferred algorithms for SNP-disease prediction.

3.1.2 Area Under the ROC

Considering now Appendix A.1 Figures 8-10, we observed similar fluctuations in AUC by correlation and number of SNP's as outlined in section §3.1.1, above. However, with respect to AUC as a holistic diagnostic metric, ADA and GBM methods clearly separate

themselves as preferred algorithms to xgBoost and deepboost (deepboost does not output necessary predicted phenotype probabilities) – especially in cases of quadratic and non-linear disease-links. Further, in all cases besides sinusoidal disease-link, boosting methods were demonstrated to be exceptionally robust diagnostic tools for SNP-disease prediction as correlation and number of SNP's being investigated grew. Concurrently, these positive attributes were only amplified as sample size grew. The boosting class of algorithms, as a whole, elicited many positive attributes with regards to SNP-disease prediction; but counter-intuitively, our simulation suggests that the simpler ADA and GBM boosting methods may be preferred over more complex algorithms *as diagnostic tools*.

3.2 Application

3.2.1 Training Diagnostics

Evaluating 100 repeated 5-fold cross validation diagnostics on the randomly selected 80% training set of 862 mice (comparable on disease outcome and features to the 20% test set), we observe from Figure 1 below that GBM and ADA elicited the lowest error and highest AUC on average, followed by xgbTree and deepboost, with xgbLinear and LASSO performing the worst on average.

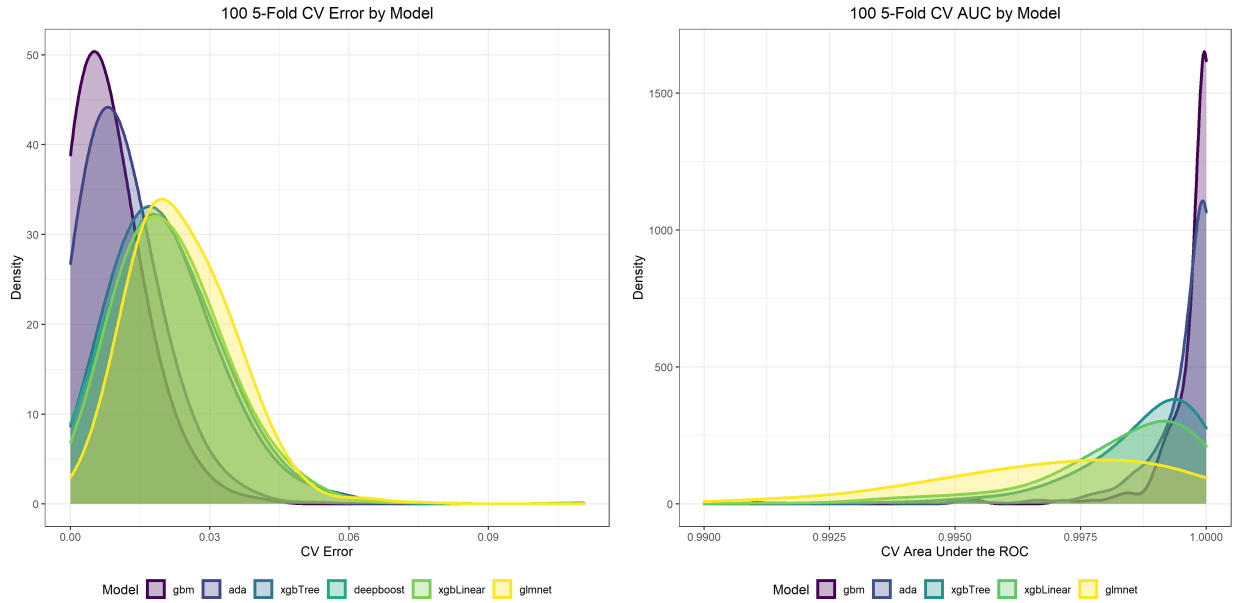

Figure 1: Distribution of 100 repeated 5-fold cross validated error rates (left) and AUC (right).

In addition, considering the numerical display in Figure 2, below, deepboost performed comparably to xgbTree in terms of error – with nearly identicial repeated 5-fold CV mean,

median, and variance. It should also be noted that in general, decision tree based methods outperformed penalized generalized linear models in SNP-disease prediction, but all methods achieved high and low cross validated mean AUC and error, respectively.

100 5-Fold CV Training Diagnostics

model	metric	mean	median	variance
gbm	auc	0.99965	1.00000	0.00000
ada	auc	0.99954	0.99986	0.00000
xgbTree	auc	0.99868	0.99906	0.00000
xgbLinear	auc	0.99828	0.99876	0.00000
glmnet	auc	0.99631	0.99713	0.00002
gbm	error	0.00828	0.00581	0.00006
ada	error	0.01094	0.01156	0.00007
xgbTree	error	0.02023	0.01744	0.00013
xgbLinear	error	0.02176	0.01744	0.00014
glmnet	error	0.02408	0.02312	0.00012
deepboost	error	0.02061	0.01744	0.00013

Figure 2: Training diagnostic results for each model by metric, mean, median, and variance.

3.2.2 Test Performance

When these models were then applied to the 'never-before-seen' testing set of 481 mice, numerical results, found in Figure 3, mirrored both simulation and training diagnostic results. AdaBoost and GBM achieved minimal error and maximum AUC, while xgBoost methods and deepboost performed slightly worse, and LASSO elicited the poorest results.

Final Testing Data Model Performance

metric	glmnet	ada	gbm	xgbLinear	xgbTree	deepboost
error	0.06237	0.02287	0.03119	0.05198	0.04782	0.05198
auc	0.98448	0.99553	0.99480	0.98661	0.98886	NA

Figure 3: Final testing data results by model and metric.

4 Discussion

This study encapsulates an exploration into the predictive capability of classical and modern 'boosted' machine learning algorithms, specifically the classification of disease phenotypes from SNP gene expression. Our findings, both in simulation and application, indicate that AdaBoost and generalized Gradient Boosting Machines outperform more complex Extreme

Boosting (xgBoost) or Deep Boosting (deepboost) methods in the context of genomics and SNP expression. These results mirror those of Li et al., who similarly found that GBM and Random Forest outperformed xgBoost in genomic prediction from SNP expression [21]. Additionally, results demonstrate emphatically that boosting decision trees ought to be favored over generalized linear models in SNP-disease prediction. While these results are insightful, and may positively influence genomic disease diagnosis, as a pilot study there exist limtitations which restrict the generalizability of the study. Below, we outline the main limitations, concluding with a brief discussion regarding future directions for the study.

4.1 Limitations

The main limitation which currently impedes the study's generalizability is derived from computational deficiency and subsequent sub-optimal algorithm tuning. AdaBoost and GBM are a less complex family of boosting algorithms with fewer parameters, and investigators' computational power to optimally tune xgBoost and deepboost over all simulations was lacking and could have potentially biased simulation results in favor of the simpler, classical boosting algorithms. Additionally, while Deep Boosting is theoretically elegant, and potentially incredibly powerful, it does not return predicted probabilities – which limits its usefulness as a diagnostic tool in clinical gene-expression studies. A re-analysis of these algorithms with improved computational ability to optimally tune the more complex algorithms is necessary to validate these results.

Additionally, with respect to the synthetic data generation, the correlation structure of each data set's SNP's was observed to be less than realistic and the lack of spread or noisiness did not reflect true GWAS exceptionally well. Further, the disease-link models (linear, quadratic, sinusoidal, and power) are not indicative of true gene expression to disease phenotype association – and other labeling techniques ought to be explored to model this relationship. In order for the simulation results to be directly transferable, improved synthetic SNP-disease data generation is necessary.

4.2 Future Directions

A natural extension of this study would be to investigate these boosting algorithms' ability to identify the "true" genetic markers associated with disease (constructed by design in the simulation). Algorithms which boost decision trees inherently quantify 'variable importance', and an inquiry into the ability of each to correctly identify "true" markers could drastically improve SNP-disease diagnoses and our holistic understanding of human health.

A Appendix

A.1 Figures

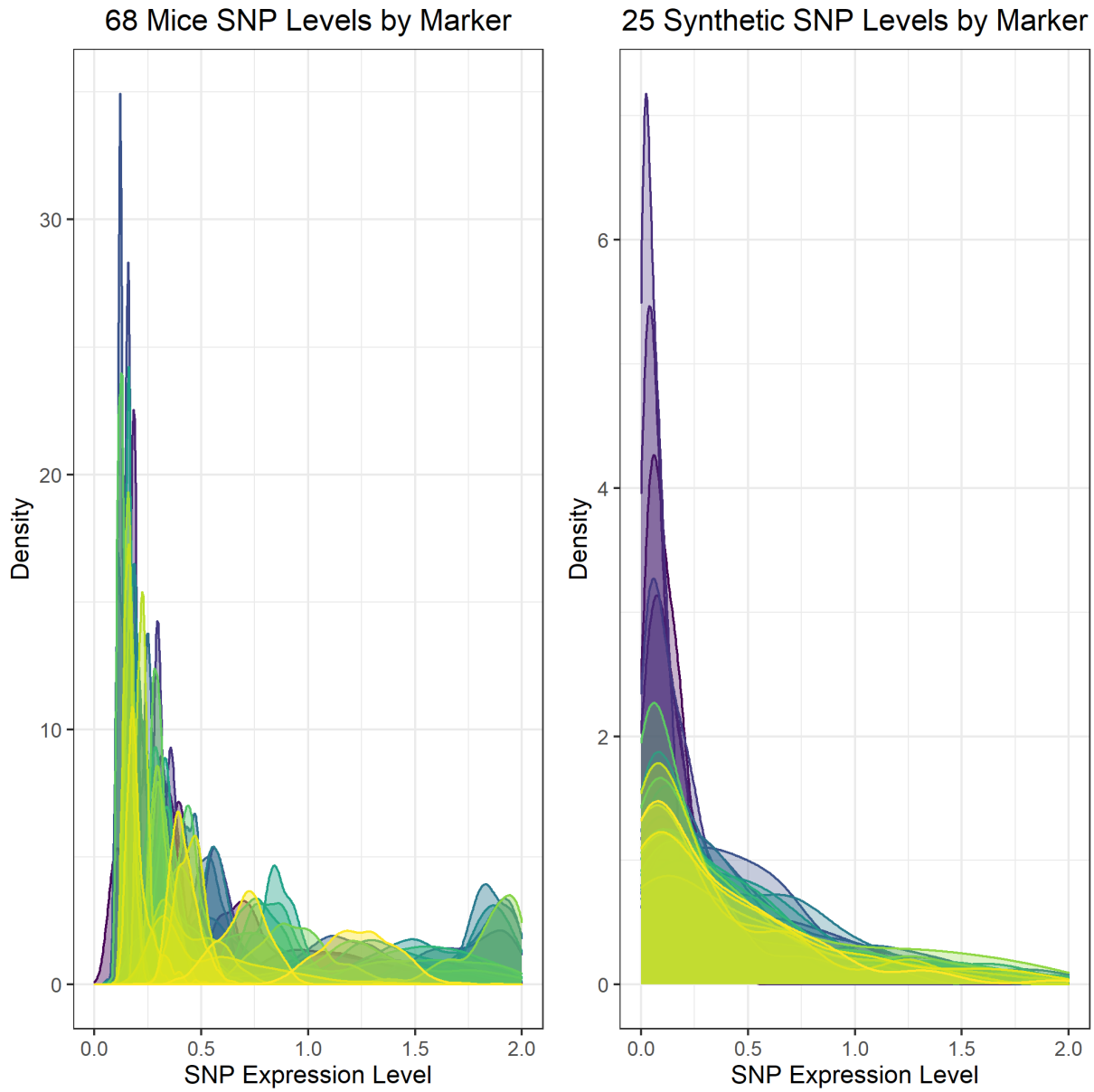

Figure 4: 68 SNP expression levels of 1077 Mice (left) versus 25 synthetic SNP expression levels of 100 observations with correlation $\rho = 0.3$ (right)

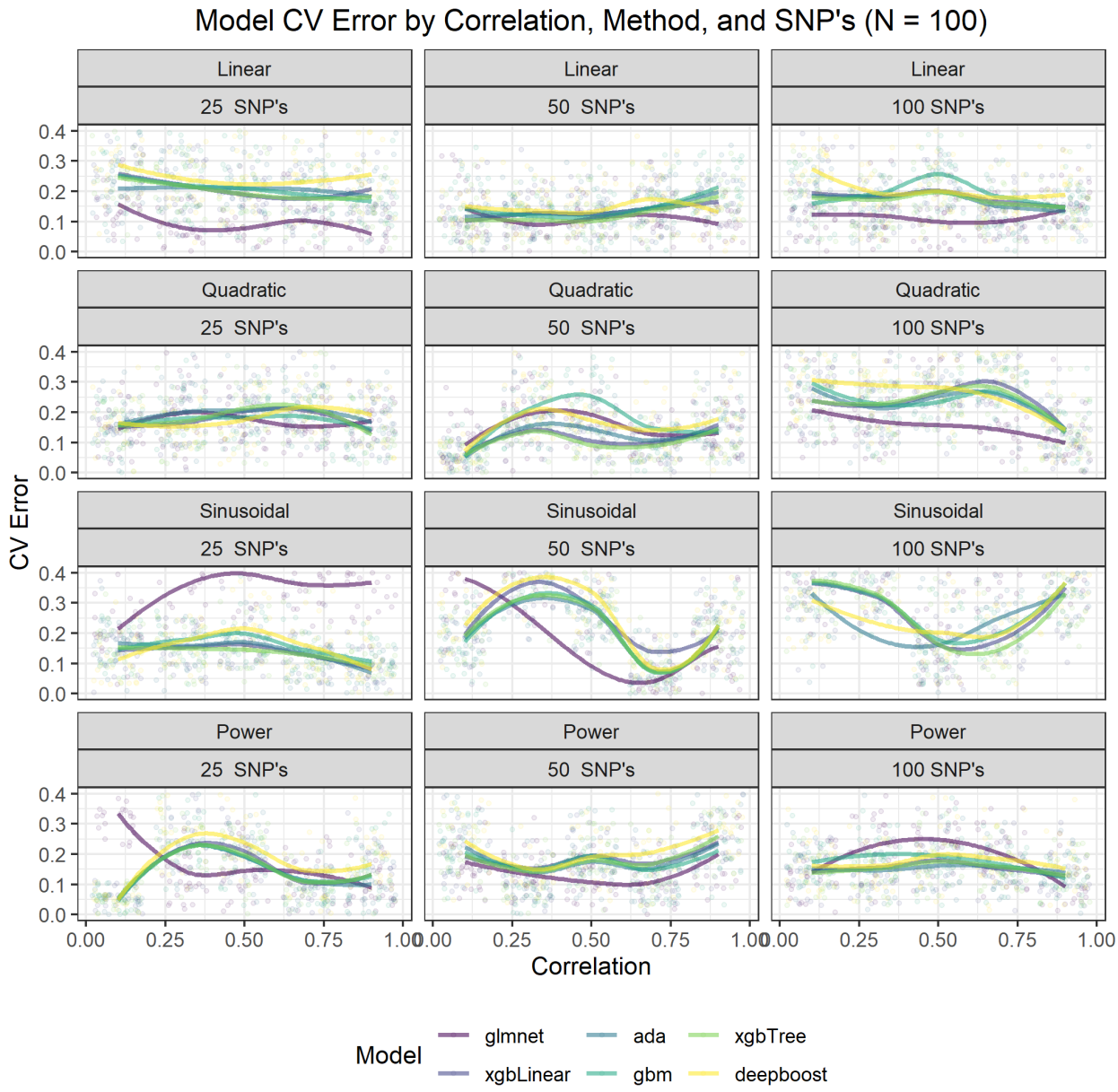

Figure 5: Repeated CV error (y-axis) by correlation (x-axis), coloured by algorithm/model, with panels describing the method of disease-link and number of SNP's. Sample size fixed at 100.

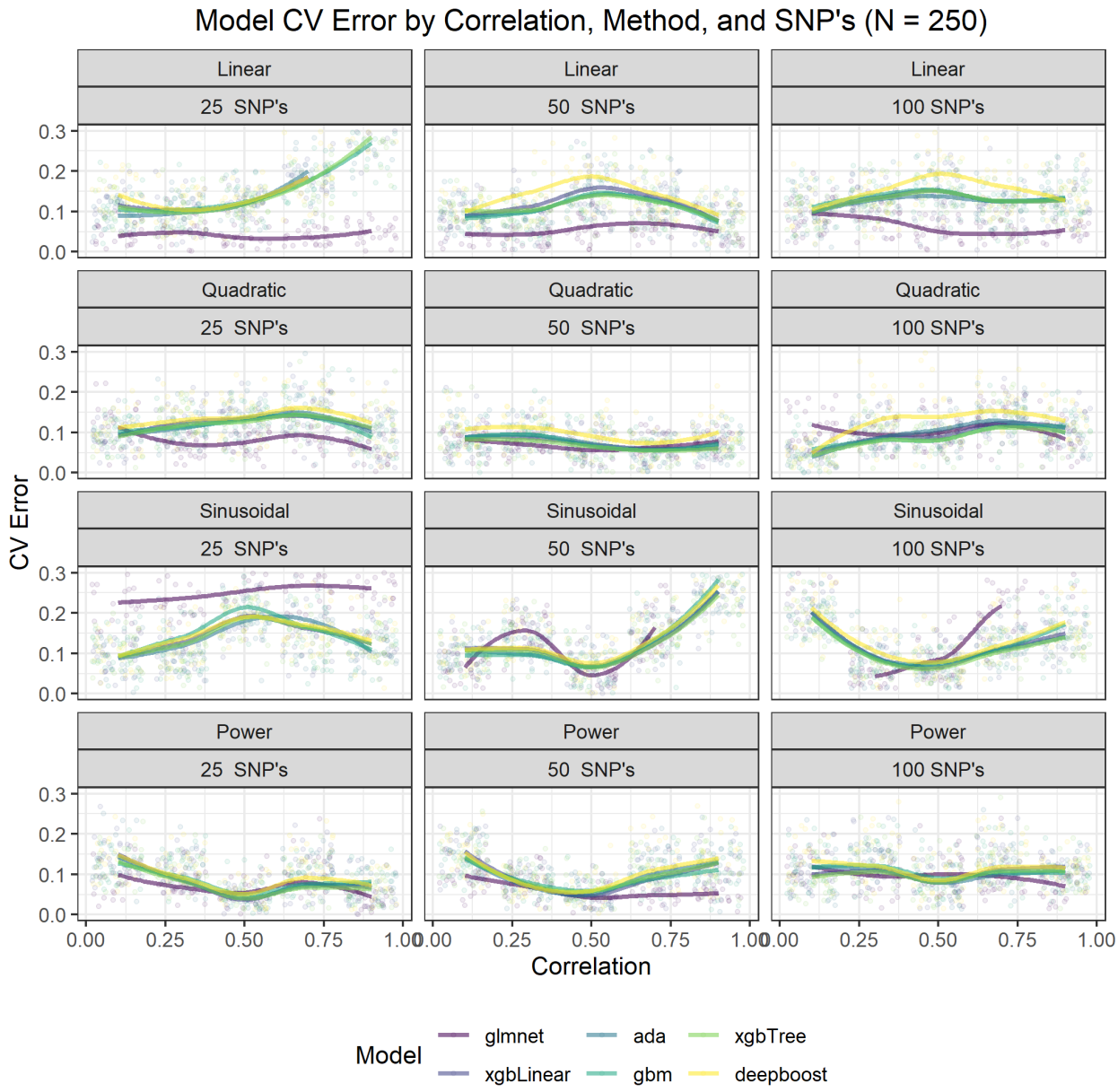

Figure 6: Repeated CV error (y-axis) by correlation (x-axis), coloured by algorithm/model, with panels describing the method of disease-link and number of SNP's. Sample size fixed at 250.

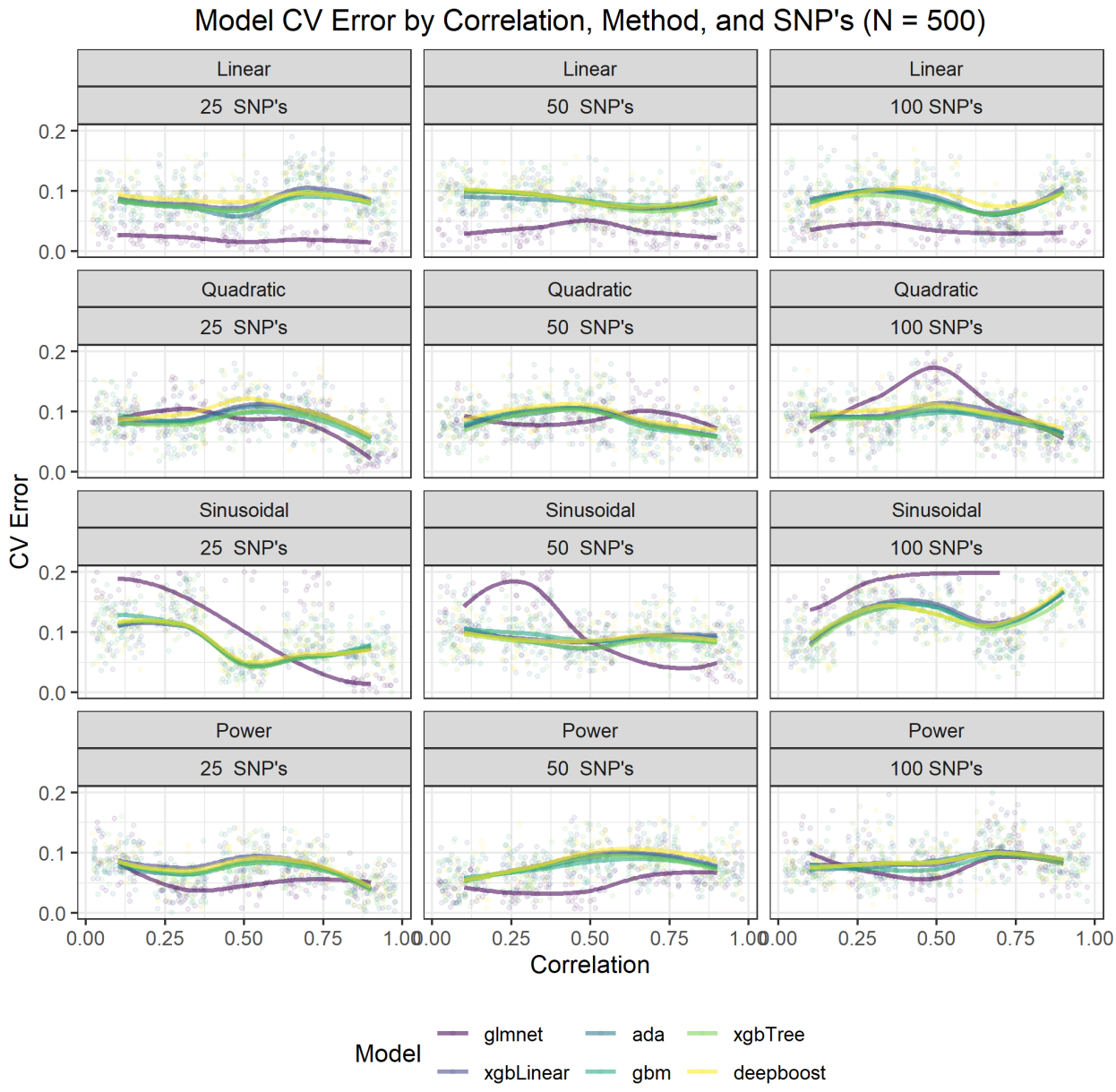

Figure 7: Repeated CV error (y-axis) by correlation (x-axis), coloured by algorithm/model, with panels describing the method of disease-link and number of SNP's. Sample size fixed at 500.

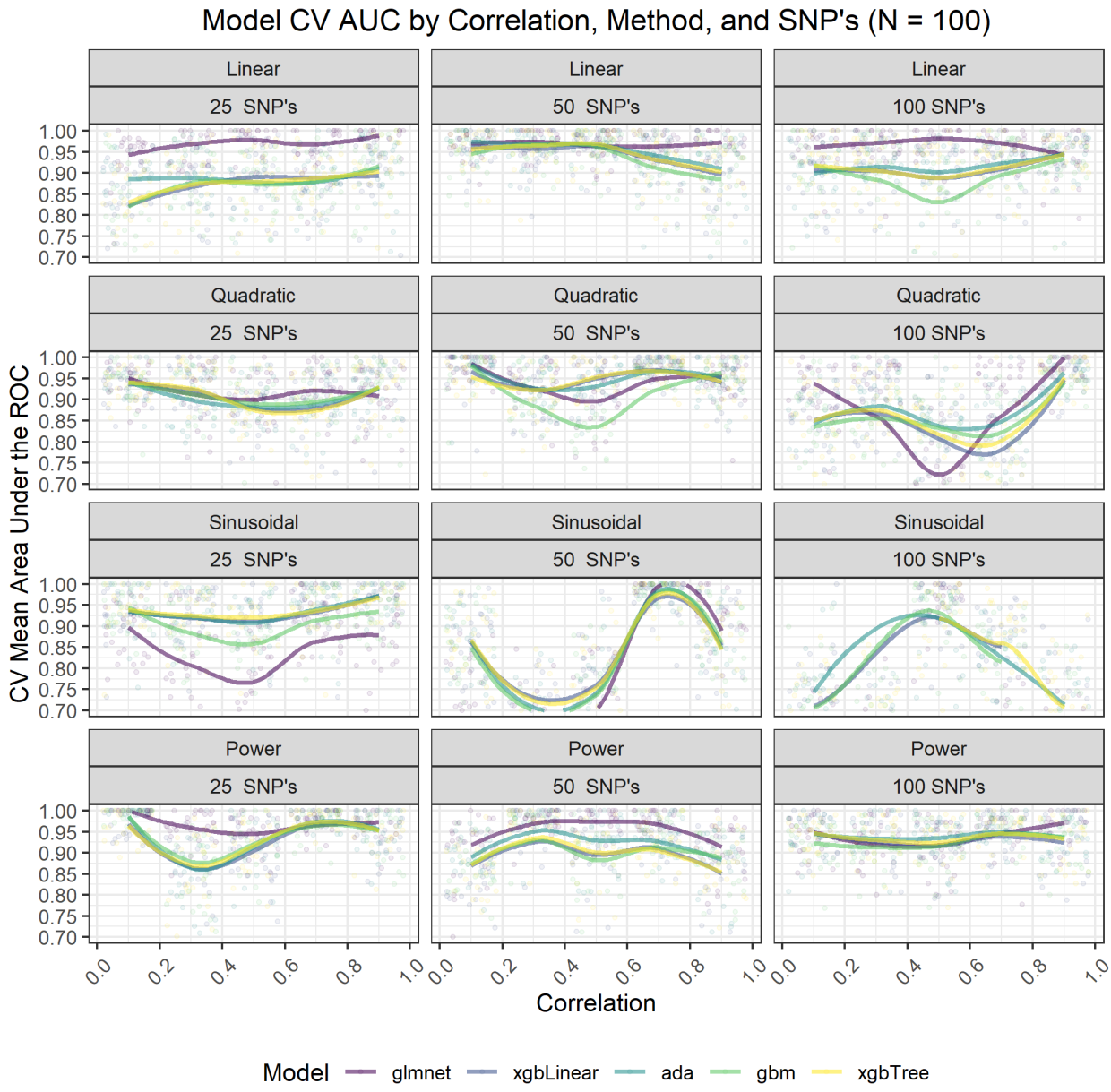

Figure 8: Repeated CV Area Under the ROC (y-axis) by correlation (x-axis), coloured by algorithm/model, with panels describing the method of disease-link and number of SNP's. Sample size fixed at 100.

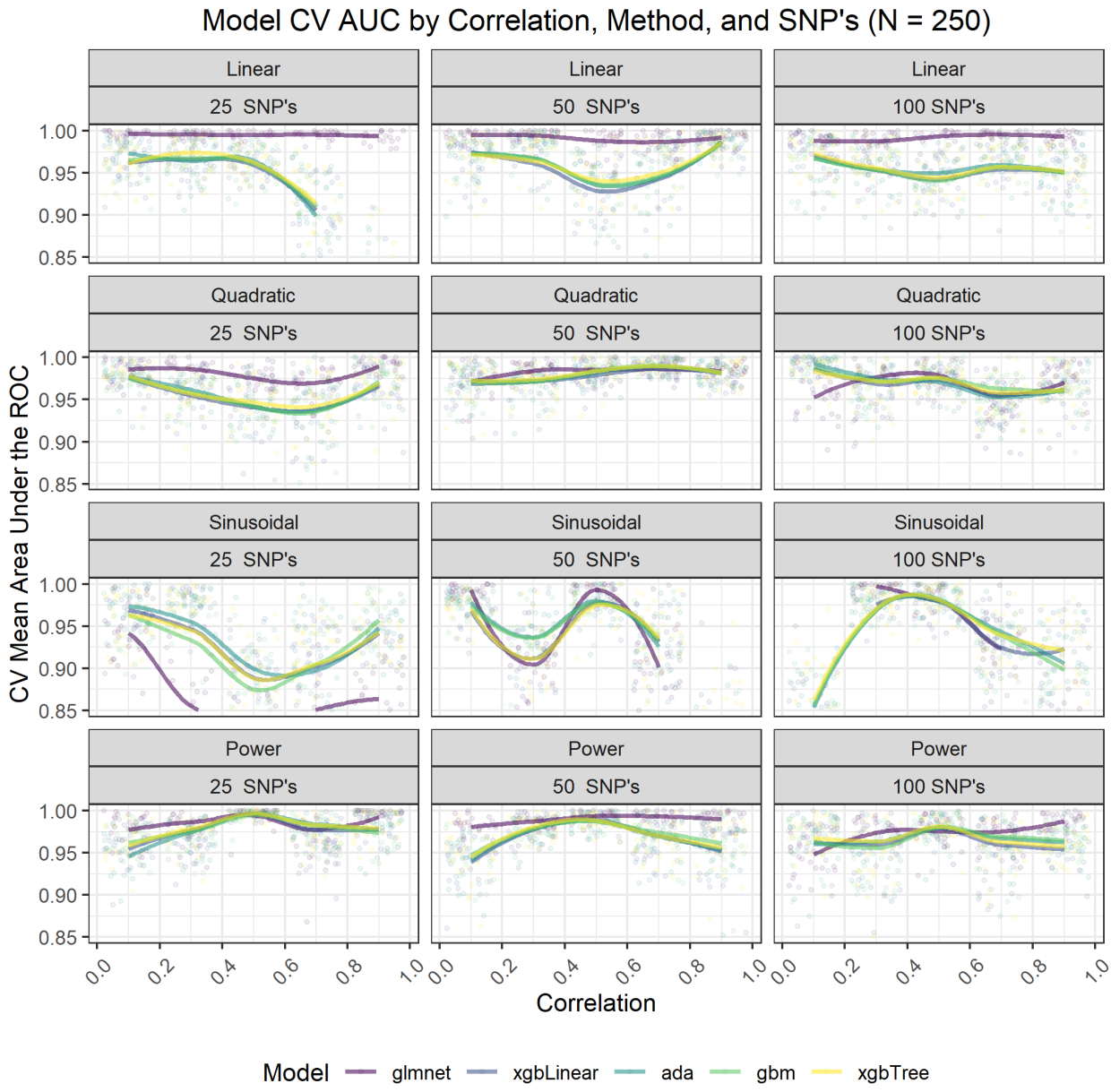

Figure 9: Repeated CV Area Under the ROC (y-axis) by correlation (x-axis), coloured by algorithm/model, with panels describing the method of disease-link and number of SNP's. Sample size fixed at 250.

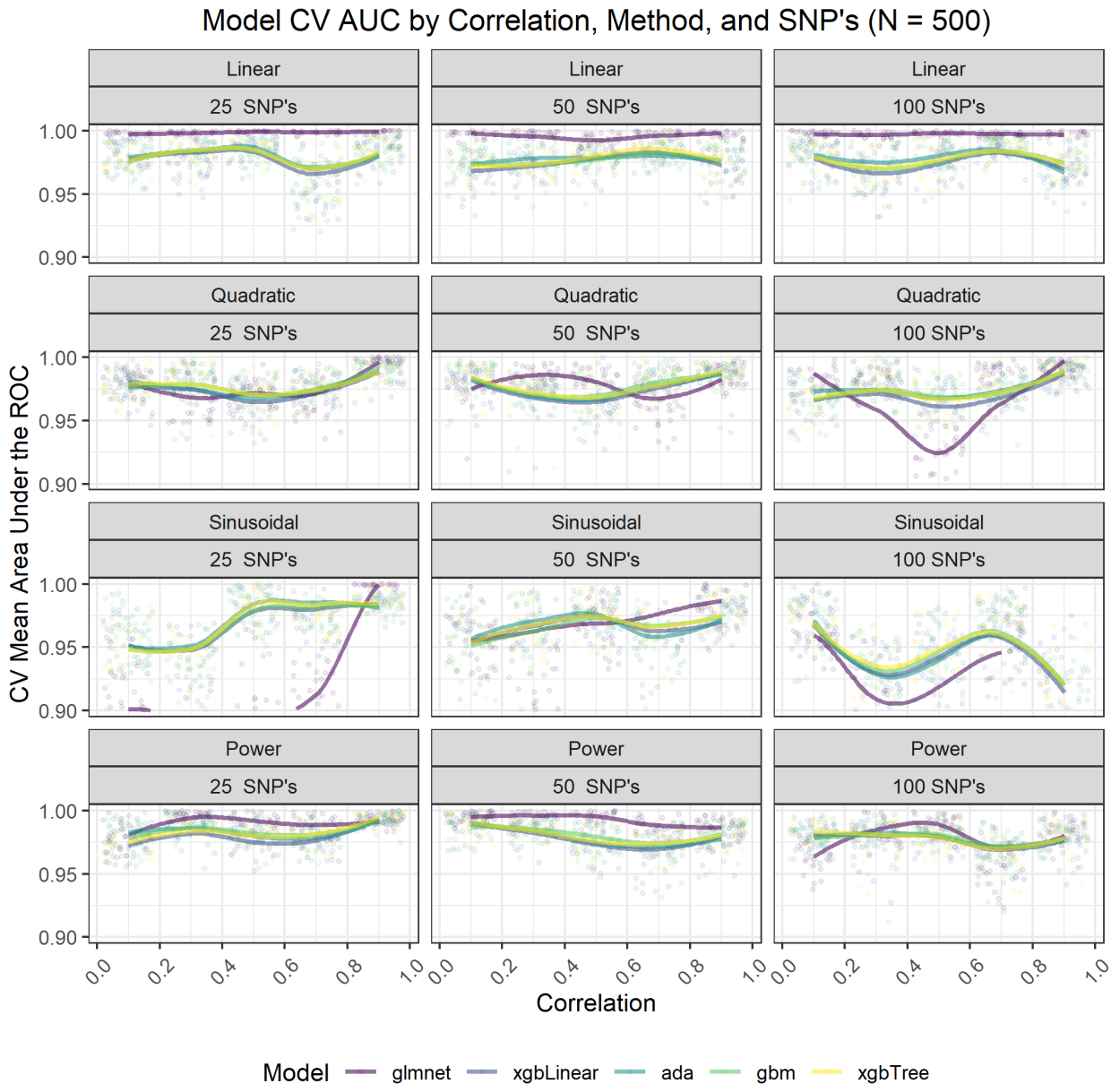

Figure 10: Repeated CV Area Under the ROC (y-axis) by correlation (x-axis), coloured by algorithm/model, with panels describing the method of disease-link and number of SNP's. Sample size fixed at 500.

References

[1] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Nature Precedings*, pages 1–1, 2010.

[2] R Bellman. Dynamic programming. 95, 1957.

[3] Martin Bengtsson, Anders Ståhlberg, Patrik Rorsman, and Mikael Kubista. Gene expression profiling in single cells from the pancreatic islets of langerhans reveals lognormal distribution of mrna levels. *Genome research*, 15(10):1388–1392, 2005.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[6] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4, 2015.

[7] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Li. *xgboost: Extreme Gradient Boosting*, 2020. R package version 1.0.0.2.

[8] Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. *Journal of Machine Learning Research*, 2014.

[9] Mark Culp, Kjell Johnson, and George Michailidis. ada: An r package for stochastic boosting. *Journal of Statistical Software*, 17(2):9, 2006.

[10] Mark Culp, Kjell Johnson, and George Michailidis. *ada: The R Package Ada for Stochastic Boosting*, 2016. R package version 2.0-5.

[11] Ziding Feng, Ross Prentice, and Sudir Srivastava. Research issues and strategies for genomic and proteomic biomarker discovery and validation: a statistical perspective. *Pharmacogenomics*, 5(6):709–719, 2004.

[12] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[14] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[15] Keith Goldfeld. *simstudy: Simulation of Study Data*, 2020. R package version 0.1.16.

[16] Brandon Greenwell, Bradley Boehmke, Jay Cunningham, and GBM Developers. *gbm: Generalized Boosted Regression Models*, 2019. R package version 2.1.5.

[17] Timothy J Griffin and Lloyd M Smith. Single-nucleotide polymorphism analysis by maldi–tof mass spectrometry. *Trends in biotechnology*, 18(2):77–84, 2000.

[18] Trevor Hastie and Junyang Qian. Glmnet vignette. *Retrieve from http://www. web. stanford. edu/˜ hastie/Papers/Glmnet_Vignette. pdf. Accessed September*, 20:2016, 2014.

[19] Eamonn Keogh and Abdullah Mueen. *Curse of Dimensionality*. Springer US, Boston, MA, 2017.

[20] Max Kuhn. *caret: Classification and Regression Training*, 2020. R package version 6.0-86.

[21] Bo Li, Nanxi Zhang, You-Gan Wang, Andrew W George, Antonio Reverter, and Yutao Li. Genomic prediction of breeding values using a subset of snps identified by three machine learning methods. *Frontiers in genetics*, 9:237, 2018.

[22] Kuang-Yu Liu, Jennifer Lin, Xiaobo Zhou, and Stephen TC Wong. Boosting alternating decision trees modeling of disease trait information. In *BMC genetics*, volume 6, page S132. Springer, 2005.

[23] Daniel Marcous and Yotam Sandbank. *deepboost: Deep Boosting Ensemble Modeling*, 2017. R package version 0.1.6.

[24] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[25] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[26] Greg Ridgeway. Generalized boosted models: A guide to the gbm package. *Update*, 1(1):2007, 2007.

[27] David Rossell et al. Gaga: a parsimonious and flexible model for differential expression analysis. *The Annals of applied statistics*, 3(3):1035–1051, 2009.

[28] Robert E Schapire. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer, 2003.

[29] Vahid Shahrezaei and Peter S Swain. Analytical distributions for stochastic gene expression. *Proceedings of the National Academy of Sciences*, 105(45):17256–17261, 2008.

[30] Yuichi Taniguchi, Paul J Choi, Gene-Wei Li, Huiyi Chen, Mohan Babu, Jeremy Hearn, Andrew Emili, and X Sunney Xie. Quantifying e. coli proteome and transcriptome with single-molecule sensitivity in single cells. *science*, 329(5991):533–538, 2010.

[31] Zhi Wei, Wei Wang, Jonathan Bradfield, Jin Li, Christopher Cardinale, Edward Frackelton, Cecilia Kim, Frank Mentch, Kristel Van Steen, Peter M Visscher, et al. Large sample size, wide variant spectrum, and advanced machine-learning technique boost risk prediction for inflammatory bowel disease. *The American Journal of Human Genetics*, 92(6):1008–1012, 2013.

[32] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019.

[33] Quin F Wills, Kenneth J Livak, Alex J Tipping, Tariq Enver, Andrew J Goldson, Darren W Sexton, and Chris Holmes. Single-cell gene expression analysis reveals genetic associations masked in whole-tissue experiments. *Nature biotechnology*, 31(8):748, 2013.