January 2011

Improving Prognostic Models In Breast Cancer With Biostatistical Analysis Of The Phosphatidyl Inositol 3-Kinase Pathway

Elliot Rapp

Improving Prognostic Models in Breast Cancer with Biostatistical Analysis of the Phosphatidyl Inositol 3-Kinase Pathway

A Thesis Submitted to the

Yale University School of Medicine

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Medicine

by

Elliot James Rapp

2011

Abstract

IMPROVING PROGNOSTIC MODELS IN BREAST CANCER WITH BIOSTATISTICAL ANALYSIS OF THE PI3-KINASE PATHWAY.

Elliot James Rapp, Jena P. Giltnane, David L. Rimm, Annette Molinaro. Department of Biostatistics, Yale School of Public Health, Yale University School of Medicine, New Haven, CT.

Our hypothesis was that prognostic models for breast cancer that incorporate both clinical variables and biomarkers in the PI3 Kinase molecular pathway will improve upon the clinical models of TNM staging and the Nottingham Prognostic Index (NPI). Our specific aim was to develop models that misclassify fewer patients than TNM and NPI with the outcome of dead of disease at ten years. Our population cohort was the YTMA49 cohort, a series of 688 samples of invasive ductal breast carcinoma collected between 1961 and 1983 by the Yale University Department of Pathology. Tissue MicroArray (TMA) analysis was performed and biomarker expression level was determined using Automated Quantitative Analysis (AQUA) technology for thirteen biomarkers in the PI3 Kinase pathway, including an overall expression level and expression levels by subcellular compartment. Eleven clinical variables were also assembled from our cohort. Exhaustively searching the multivariate space, we used logistic regression to predict our outcome of dead of disease at ten years. Validation was performed using Leave One Out Cross Validation (LOOCV). Misclassification estimates provided the means to compare different models, with lower misclassification estimates indicating superior models. Confidence

intervals were constructed using bootstrapping with one thousand iterations. We developed a helper computer program named Combination Magic to enable us to develop sophisticated models that included both interactions between variables and transformations of variables (e.g. logarithm).

Overall our best univariate models were NPI (misclassification estimate (ME): 0.326, confidence interval (CI): 0.292 to 0.359), Nodal status (ME: 0.353, CI: 0.322 to 0.493), and TNM (ME: 0.367, CI: 0.313 to 0.447). Our best univariate models from the PI3 Kinase biomarkers were FOX01_NU (ME: 0.369, CI: 0.336 to 0.415), AKT1_TM (ME: 0.373, CI: 0.335 to 0.412), and PI3Kp110_TM (ME: 0.377, CI: 0.343 to 0.431). Our best bivariate models were pTumor*PathER (ME: 0.328, CI: 0.308 to 0.443), pNode + NuGrade (ME: 0.333, CI: 0.305 to 0.434), and AKT1_NN + Fox01_NU (ME: 0.338, CI: 0.307 to 0.391). Our best trivariate models were pTumor + mTOR_NN + PI3Kp110_TM + pTumor*PI3Kp110_TM (ME: 0.296, CI: 0.273 to 0.375), pTumor + AKT1_NU + Fox01_NU + pTumor*AKT1_NU (ME: 0.298, CI: 0.275 to 0.38), and pTumor + mTOR_TM + PI3Kp110_TM + pTumor*PI3Kp110_TM (ME: 0.299, CI: 0.276 to 0.378). Our best multi-variate model was Fox01_NU + AKT1_NU + mTOR_MB + p70S6K_NU + AVG_BCL2_TM + Fox01_NU*AKT1_NU*mTOR_MB (ME: 0.295, CI: 0.274 to 0.393). None of these models was statistically superior to the clinical models of TNM and NPI.

Acknowledgements

My mentor Dr. Annette Molinaro has been a tremendous pillar of support throughout this project and I want to thank her for always patiently explaining the intricacies of our statistical methodology, helping me overcome the computing hiccups that invariably happen in a project that involves tens of millions of computations, and always making time to meet with me. Despite the demands of her busy schedule, including teaching, research and a new addition to her family, I always found her with a smile on her face when I walked into her office. She is fantastic and a true inspiration.

Dr. David Rimm, my co-mentor on the bench side of this project, was equally important to the completion of this work. He provided us with all of the data we needed to perform our analysis and graciously offered his lab resources to train me in the AQUA methodology, thus greatly increasing my understanding of the underlying technology we used to quantify our biomarkers. Despite his incredible knowledge and expertise, Dr. Rimm is as down-to-earth as can be, and would win my award for "PI you most want to get a beer with".

Dr. Jena Giltnane was my partner in crime for this work. Using AQUA, she quantified all of the biomarkers in the PI3K pathway, gathered the clinical data, and patiently explained the underlying biological pathways. She was a champ to work with and always brightened the room with her presence. I wish her the best as she finishes her pathology residency at Vanderbilt.

Table of Contents

Introduction

Although many advances have been made in breast cancer treatment over the last few decades, it remains a deadly disease with significant financial, health, and emotional costs for breast cancer patients and survivors. Of particular concern for both doctors and patients is likelihood of recurrence after resection of the primary tumor. This likelihood can have an immediate impact on choice of treatment, including the type of medical treatment and the type of surgical resection, from lumpectomy to simple mastectomy to radical mastectomy.

The likelihood of recurrence also carries an emotional toll long after treatment of the primary tumor is completed. Many patients live in fear of recurrence. Patients may choose a more aggressive chemotherapy treatment than necessary, which carries its own health risks and litany of side effects. Patients with a particularly high risk profile may even opt for prophylactic mastectomy of a healthy breast.

Clinical models have been developed to assess the risk of metastasis in breast cancer and are widely used in clinical practice, despite the fact that their utility continues to be a matter of debate. The best known are TNM staging and the Nottingham Prognostic Index (NPI). Each has been validated in numerous clinical trials.

TNM staging is based on tumor size, number of positive lymph nodes, and the presence of metastasis. The TNM classification of breast cancer was updated in 2002 and again in 2009. As our work was performed prior to the

2009 definition, the 2002 definition was used for this analysis. Although widely used historically, TNM staging of breast cancer has been criticized as having limited utility in actual clinical practice. As far back as 1992 Barr and Baum called for its removal from clinical decision making, arguing that it ignores many factors relevant to both surgical treatment and prognosis, and that factors it does include are difficult to reliably assess clinically, with unacceptably high false positive and negative rates (1).

More recently, Benson continues the criticism of the clinical utility of TNM staging in breast cancer (2). He notes that TNM staging was developed at a time when the pathological model of cancer metastasis was thought to happen in an orderly, Halstedian fashion, with cancer spreading in a logical manner from its site of origin to local lymph nodes to distant sites of metastasis. However, small tumors with aggressive hematogenous spread do not follow this model, and tend to be more aggressive than larger tumors with lymphatic spread. Thus, TNM staging is particularly unhelpful for this type of tumor, motivating the need for a more helpful alternative.

The NPI prognostic model is based on tumor size, number of positive lymph nodes, and the histologic grade. It yields a continuous number that falls into one of six prognostic groups, from Excellent Prognostic Group (EPG) to Very poor Prognostic Group (VPG). NPI is the only breast cancer staging model with prospective validation both intra and inter-center (3). It differs from TNM in that it includes histologic grade, yet like TNM it does not incorporate biomarkers. Thus, neither TNM staging nor the Nottingham Prognostic Index

incorporate the growing body of knowledge regarding the significance of different subtypes of breast cancer and the importance of metabolic and signaling pathways in breast cancer growth and metastasis.

With advances in our understanding of the molecular pathways involved in breast cancer, there has been renewed interest in developing prognostic models that include biomarkers to overcome limitations of TNM staging and the Nottingham Prognostic Index and more accurately predict metastasis and/or recurrence in breast cancer. Significantly, laboratory tools such as Automated Quantitative Analysis (HistoRX, New Haven, Connecticut) also allow us to more accurately quantify expression levels of biomarkers in various pathways significant in cancer, improving the accuracy and reproducibility of models using biomarkers.

More accurate prognostic models will provide value to both physicians and patients. Biomarker analysis can also provide important information about the likely efficacy of various drugs that are targeted at different molecular pathways active in breast cancer. A notable example is the use of the pharmaceutical Trastuzumab to target the HER2/neu receptor, which has been widely used in clinical practice since its FDA approval in 1998 (4).

After mutations of the p53 tumor suppressor gene, mutations within the Phosphatidyl Inositol-3 Kinase (PI3 Kinase or PI3K) pathway are the most common mutations leading to human cancer (5). Recent work has shown the particular significance of PI3 Kinase mutations in human breast cancer (6). Given the need for more accurate prognostic models in breast cancer, the

likelihood that inclusion of biomarkers in prognostic models can improve upon the existing clinical models, and the importance of the PI3K pathway in breast cancer, we chose to evaluate whether a search of the covariate space of various messengers in the PI3 Kinase pathway and clinical variables can offer improved prognostic models when compared to the clinical gold standards of TNM and NPI.

Specific Aims of Thesis

Aim 1

Determine whether an exhaustive search of the covariate space of the PI3 Kinase molecular pathway and clinical variables in metastatic breast cancer can improve prognostic models over the existing clinical standards of TNM staging and the Nottingham Prognostic Index (NPI) with the outcome of dead of disease at ten years. Evaluate whether interactions between different biomarkers in the PI3 Kinase pathway and logarithmic values can improve upon the prognostic models previously identified.

Aim2

Develop a reliable methodology to create and evaluate prognostic models with an arbitrary number of interactions, sub-interactions, and custom terms. In addition, structure our computational analysis to allow us to both exhaustively search the model space and selectively create models by hand.

Methods

Statistical Platform

All computations were performed using The R Project for Statistical Computing, an open-source statistical language and environment freely available on the Internet (7).

Tumor Cohort

The YTMA49 cohort has been previously described (8). Briefly, it consists of 688 samples of invasive ductal breast carcinoma collected between 1961 and 1983 from the Yale University Department of Pathology archives. The mean and median age of diagnosis were 58.1 and 58.0, respectively. The mean and median follow-up times were 12.8 and 8.9 years, respectively. The cohort contains approximately half node-positive and half node-negative specimens.

Data Acquisition

Data acquisition was performed by Jena Giltnane, at the time a MD/PhD candidate at Yale University School of Medicine. She analyzed 539 metastatic breast cancer biopsy cores from the YTMA49 cohort using Automated Quantitative Analysis (described below). In addition to quantifying expression levels of thirteen biomarkers in the PI3 Kinase pathway (see Table One), she assembled eleven clinical variables (see Table Two). Together, these 24 variables were included as the covariates for model building. However, as noted below, each biomarker was further characterized by its expression level in three subcellular compartments. Thus, there were a total of 63 possible inputs to our model generation (52 biomarkers and 11 clinical variables).

AQUA Analysis of the PI3 Kinase Pathway

Automated Quantitative Analysis (HistoRX, New Haven, Connecticut), known as AQUA, has been previously described (9). Traditional scoring of protein expression performed by pathologists using immunohistochemistry and visual inspection of tumor slides is limited by inter-operator variability and lack of reproducibility. AQUA technology assigns a value from 0 to 255 to represent the level of biomarker expression, with a higher number indicating greater expression. Its increased granularity of expression levels and high reproducibility when compared to traditional methods is designed to allow for more accurate prognostic models.

When performing AQUA quantification, the technician distinguishes areas of tumor from stromal elements by staining with antikeratin and creating an epithelial tumor mask. After the operator visual sets an intensity threshold to distinguish between cancerous and non-cancerous areas, the AQUA software defines each area as “on” (tumor) or “off” (non-tumor). By convention, TM (“Tumor Mask”) describes the overall expression level in cancerous cells.

In addition, AQUA allows quantification of protein expression by subcellular compartmentalization. NU (“Nuclear”) describes the expression level in the nuclear compartment. MB (“Membrane”) describes the expression level in the cellular membrane. NN (“Non-Nuclear”) describes the expression level in the non-nuclear, non-membranous portion of the cell. The ability to localize biomarkers in the PI3 Kinase pathway by subcellular compartment is

designed to allow for improved prognostic models by more precisely describing the metabolic activity of the pathway.

Model Search

One of the challenges of our analysis was model selection. One possibility is to use artificial intelligence algorithms to search the model space and use pruning techniques to eliminate unfavorable portions of the covariate space. This is potentially advantageous given limited computing resources. However, without deep understanding of the domain space, it is difficult to accurately predict which portions of the covariate space are unfavorable for analysis. Another possibility is to use classic Classification and Regression Trees (CART) (10) to determine good variables on which to split and then construct models by hand. Although this lessens the dependence of model quality on the researcher's domain expertise, it remains prone to error. The most foolproof method, given sufficient computing resources, is to exhaustively search the variable space. With the availability of two high-performance clusters (described below) on which to perform our analysis, we chose the exhaustive search option.

With exhaustive search of the covariate space, it becomes imperative to accurately create all possible combinations of variables for models of varying complexity (e.g., univariate, bivariate, trivariate, and so forth). Given combinatorial explosion, this can be a time-consuming task. In addition, we desired the ability to create specialized runs with models that included straightforward interactions of the form "Y ~ A*B*C", where Y is the outcome

and A, B, and C are inputs, sub-interactions of the form "Y ~ A + B+ C + A*B", where "A*B" defines an interaction between A and B in addition to their independent effects, and custom and/or transformed terms (e.g., logarithmic terms, such as Y ~ A + log(B/C)).

Logistic Regression Analysis and Misclassification

Given that our goal was to improve prognostic models of breast cancer, we chose dead or alive of disease as our outcome. An important consideration was the appropriate time interval. Shorter time intervals would classify patients with later recurrence as having "survived" breast cancer. Longer time intervals would compromise the analysis by introducing the confounding nature of comorbidities that are common in older patients, as well as increasing the number of patients that did not have sufficient follow up. A ten year time interval was chosen as a compromise. Thus, our outcome was death due to breast cancer within ten years of the initial diagnosis.

As we needed a statistical methodology that would predict this binary outcome, we chose logistic regression. The logistic regression computation was performed using R's glm ("generalized linear model") function.

Leave One Out Cross Validation

In order to estimate prediction error of our models, we chose Leave One Out Cross Validation (LOOCV). In this methodology, each patient is left out of the training set in turn and used in the test set. A logistic regression analysis is performed on the training set. Maintaining the parameters of the resulting model, if it correctly predicts the outcome (i.e., alive or dead at ten years) for

the patient left out (i.e., the single patient in the test set), then its classification is correct. If it incorrectly predicts this outcome, then its classification is incorrect. After all n-iterations of LOOCV, there are as many classifications as patients. By dividing the incorrect classifications by the total number of classifications, we arrive at a misclassification estimate. Our goal in this research project was to minimize the misclassification estimate, thereby improving our ability to accurately predict alive/dead status at ten years.

Note that due to missing data values, not every patient will be included in the analysis when creating a misclassification estimate for a given model. Thus, as a measure of quality, we counted the number of successful predictions that were used to construct each misclassification estimate (the denominator of the misclassification estimate).

In addition to misclassification, we computed Area Under the Curve (AUC) from ROC curves (Receiver Operator Curves). The ROC curves were constructed using R's rcorr.cens function from the Hmisc library (11).

In order to format the data in a manner palatable to R, we were required to make a number of changes. For example, R does not allow numerals at the start of column or row names in a data table. Another potential issue was "factor" variables. These include classification variables (e.g., TumorType could be Ductal or Lobular) and ordinal variables (pathologist-scored biomarker expression level scored as a 0, 1, 2 or 3). In R, the former group is automatically recognized, whereas the latter group must

be explicitly designated as a factor; otherwise, it will be interpreted as a continuous variable.

Another consideration was the alphabetical order of the factor names. This is due to the fact that R designates whichever factor is first alphabetically as the baseline. In order to set the factor with the largest number of instances as baseline, we would programmatically recode the values when necessary, using 0, 1, 2 … n to represent the various categories in order of prevalence, and then explicitly designate each relevant variable as a factor to prevent its interpretation as a continuous variable.

Bootstrapping and Confidence Intervals

Next, we needed to develop confidence intervals for our models, allowing us to compare two models and determine whether their misclassification estimates were statistically different. A common method for forming confidence intervals is via bootstrapping, which entails building training sets by sampling the data with replacement (12). This did not change the total number of patients in the training set, but it did mean that patients could be included two or more times, whereas other patients would not be included at all. By repeating this process 1000 times, it was possible to construct a confidence interval by evaluating the 2.5 and 97.5 percentiles.

Cluster Runs

Runs were performed on two different high-performance computing clusters maintained by the Yale University Life Sciences Computing Center, supported by NIH Grant RR19895. The first, BulldogC, contains one hundred

and thirty compute nodes. Each consists of dual 3.2 Ghz EM64T Xeon, 64-bit processors with eight GB of available RAM. The second, BulldogI, contains one hundred and seventy compute nodes. Each consists of two dual core 3.0 GHz Xeon 5160 processors with sixteen GB of available RAM. Runs varied greatly in complexity and required anywhere from a few hours to a week of processing time spread over one to one hundred and eighty processors.

Given the enormous size and complexity of the runs, gracefully recovering from errors and having the ability to re-run selective inputs became a priority. As noted earlier, we gained the ability to reproduce parts of the run by extracting model creation from the analysis code and passing discrete units of work into the R analysis function (represented by our input files).

R was particularly sensitive to some data configurations. This was especially true in the case of factor variables. For example, in cases of a patient with a unique value for a factor variable, a R run-time error would occur when the patient was left out of model creation and then used for prediction during LOOCV. Such a failure would cause the entire input file to fail and it was difficult to pinpoint the responsible formula. To gracefully recover from these errors, we wrapped with R's tryCatch function the code for generating the logistic regression model, making the prediction, and computing the Area Under the Curve (AUC).

Results

Univariate Results

Univariate	Mis.Est.	AUC	n	Mean	0.025	0.975
NPI	0.326	0.644	473	0.326	0.292	0.359
~pNodal Stage	0.353	0.597	539	0.372	0.322	0.493
TNM	0.367	0.609		0.367	0.313	0.447
~Metastasis Stage	0.368	0.537	536	0.367	0.334	0.398
~Fox01_NU	0.369	0.524	434	0.375	0.336	0.415
~AKT1_TM	0.373	0.5	415	0.374	0.335	0.412
~PI3Kp110_TM	0.377	0.539	403	0.386	0.343	0.431
~mTOR_TM	0.382	0.5	429	0.383	0.346	0.419
~NFkB_TM	0.383	0.5	439	0.389	0.35	0.433
~HER2_MB	0.385	0.524	535	0.391	0.357	0.425

Figure 1: Selected Univariate Results

Not surprisingly, the clinical models of NPI and TNM were superior to any single biomarker in the PI3 Kinase pathway. This is likely because each of these clinical models incorporates multiple data points into a single marker of disease severity. Interestingly, nodal status (the presence of metastasis in adjacent lymph nodes) was by itself slightly (but not significantly) more predictive than TNM overall, followed closely by whether or not the disease had metastasized distantly. Of the PI3 Kinase variables, the nuclear localization of Fox01 and the overall expression of AKT1 were most predictive.

Bivariate Results

Bivariate	Mis.Est.	AUC	n	Mean	0.025	0.975
pTumor * PathER	0.328	0.632	472	0.357	0.308	0.443
pNode + NuGrade	0.333	0.645	502	0.357	0.305	0.434
AKT1_NN + Fox01_NU	0.338	0.54	349	0.349	0.307	0.391

Figure 2: Selected Bivariate Results

Although our top three bivariate models all approached the univariate NPI model with misclassifications of 0.328, 0.333, and 0.338, respectively, none of them were quite able to match its 0.326 misclassification estimate. Furthermore, whereas the 95% confidence interval for misclassification by NPI was only 0.067 wide, the confidence intervals for two of our two top three bivariate models were considerably wider, indicating a greater likelihood that in future analysis our favorable results would not be replicated on independent data sets

Trivariate Results

Trivariate	Mis.Est.	AUC	n	Mean	0.025	0.975
pTumor + mTOR_NN + PI3Kp110_TM + pTumor*PI3Kp110_TM	0.296	0.648	311	0.322	0.273	0.375
pTumor + AKT1_NU + Fox01_NU + pTumor*AKT1_NU	0.298	0.620	332	0.323	0.275	0.38

pTumor + mTOR_TM + PI3Kp110_TM + pTumor*PI3Kp110 _TM	0.299	0.648	311	0.326	0.276	0.378

Figure 3: Selected Trivariate Results

With the additional information provided by a third variable, our top three trivariate results showed considerable improvement over our best bivariate results. They each had an improved misclassification estimate over NPI. However, they also had a relatively wide 95% confidence interval of approximately 0.1. Thus, the 95% confidence intervals of each overlapped with the 95% confidence intervals of the clinical models of TNM and NPI, and as a result none of the results were statistically significant.

Multi-Variate Results

	Mis. Est.	AUC	n	Mean	0.025	0.975
Fox01_NU + AKT1_NU + mTOR_MB + p70S6K_NU + AVG_BCL2_TM + Fox01_NU*AKT1_NU*mT OR_MB	0.295	0.587	285	0.33	0.274	0.393
Fox01_NU + AKT1_NU + mTOR_MB + AVG_BCL2_TM + p70S6K_NN + Fox01_NU*mTOR_MB*p7 0S6K_NN	0.302	0.574	285	0.33	0.277	0.391
Fox01_NU + AKT1_NU + mTOR_MB + AKT2_NN + AVG_BCL2_TM + p70S6K_NN + Fox01_NU*AKT1_NU*mT OR_MB	0.295	0.593	285	0.331	0.273	0.401
Fox01_NU + AKT1_NU + mTOR_MB + cmyc_NU +	0.297	0.627	195	0.333	0.266	0.403

AVG_BCL2_TM + p70S6K_NN + AKT1_NU*p70S6K_NN						

Figure 4: Selected Multivariate Results

Our top multivariate (i.e., greater than trivariate) results did not show improvement over our best trivariate models. Misclassification estimates were similar but the confidence intervals widened.

Discussion

Significance of Study Results

Our results were not significant when compared to the commonly used clinical models of TNM and NPI. There are several possible reasons for this, including issues with missing data values, incomplete patient follow up in the YTMA49 cohort, choice of statistical methodology, and the significance of other biological pathways besides the PI3 Kinase pathway in breast cancer.

First, there was a moderate amount of missing data, including clinical variables that were not available and AQUA variables that could not be computed due to inadequate tumor cores. Logistic regression drops patients that do not have all values for all variables in the model, which leads to reduced accuracy of the model.

Furthermore, patients with missing data values were also not available when creating our confidence intervals with bootstrapping. With a smaller number of patients, this may have caused our confidence intervals to be wider than they would have been otherwise, and could have sacrificed statistical

significance in some cases. Although it carries its own risks, imputing data may have improved our model accuracy and reduced our confidence intervals by making these patients available for regression analysis.

Second, there was incomplete and inconsistent follow up of patients in the YTMA49 cohort. This may have led to patients that were incorrectly classified as dead of disease that in actuality died of another cause. It may also have led to patients that were incorrectly classified as dying of another cause. In addition, some patients were lost to follow up altogether.

Third, our results may also have suffered from our choice of outcome. The use of dead-of-disease at ten years arbitrarily separates a patient who has recurrence leading to death at nine years from one having the same outcome at ten years, even though these patients are effectively the same from a survival perspective. Instead, choosing random forests with survival trees may have improved our results by eliminating the arbitrary cutoff of ten years (13, 14). Another option would be to use dead of disease at 15 years instead of 10 years, given that the longer timeframe should eliminate the majority of late recurrences. Separately, removing patients older than 80 would eliminate the confounding nature of the significant comorbidities in this elderly group of patients.

Perhaps most significantly, since our study was designed, the importance of other pathways in breast cancer besides the PI3 Kinase pathway has become increasingly apparent. This includes the p53 pathway and several others as well (15). In some cases, multiple pathways may operate independently. In

other cases, the pathways may have a dependency on each other. For this reason, despite the tradeoff of a smaller cohort of patients, our analysis may have benefited from a focus only on the HER2-positive cases, as these are the cases for which the activity of the PI3-Kinase pathway has the most importance.

Obviously, future studies of the potential of biomarker analysis to improve breast cancer prognosis would likely also benefit from the inclusion of other biological pathways that are significant in breast cancer.

Statistical Methodology

Commonly, two statistical methodologies are used to create prognostic models to predict a binary outcome in human disease. These are logistic regression and artificial neural network models (16). Other methodologies include k-Nearest Neighbors, Linear Discriminant Analysis, and Classification and Regression Trees (10).

Significant advantages of logistic regression are that the methodology is well established and the coefficients of the models have intuitive clinical interpretations (17). This allows us to compare the relative importance of various actors within the dysregulated PI3 Kinase pathway in breast cancer. Furthermore, logistic regression has been used in previous studies attempting to evaluate clinical models for breast cancer diagnosis (18, 19), and we were interested in comparing molecular models developed by this methodology to the clinical models used in practice.

Artificial neural networks are another viable option. They have the advantage of not assuming a linear relationship between the model inputs and its outcome. However, they are prone to over-fitting. Additionally, their "black box" nature makes it impossible to reliably compare the relative importance of the various inputs in the model (17). For the above reasons, we chose to use logistic regression over artificial neural networks in our analysis.

However, in practice artificial neural networks have often performed well in elucidating previously unforeseen predictors in prognostic studies. It would be instructive in a future study to compare the performance of artificial neural networks to the performance of logistic regression with regard to predicting survival in the YTMA49 cohort.

Cross Validation

There are many methods of performing cross validation. The three most commonly used are v-fold Cross Validation, Leave One Out Cross Validation, and Monte Carlo Cross Validation.

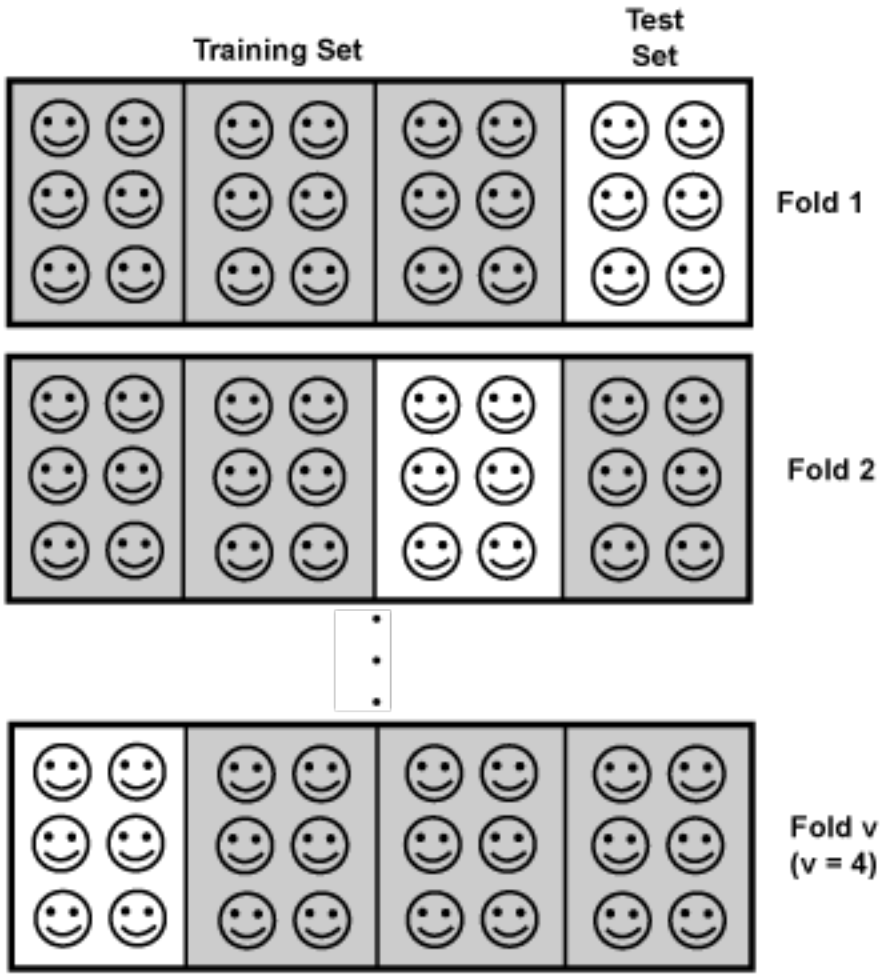

Figure 5: v-fold Cross Validation

In v-fold Cross Validation, the cohort is split into v equal partitions. v-1 of these partitions are used for the training set, while the vth partition is used for the test set. In the next iteration, a different partition is used for the test set, while the remaining partitions are again used for the training set, and the process repeats. Thus, each partition is used as the test set exactly once and is included in the training set v-1 times.

Leave One Out Cross Validation (LOOCV) is the most extreme example of v-fold cross validation, where n is the size of the sample and v = n. This means

that the regression analysis is performed n times, the size of the training set is n-1 patients, the size of the test set is one patient, and each patient is in the test set exactly once. Its thoroughness results in a smaller bias than lesser forms of v-fold cross validation. A tradeoff is that it is the most computationally intensive form of v-old cross validation.

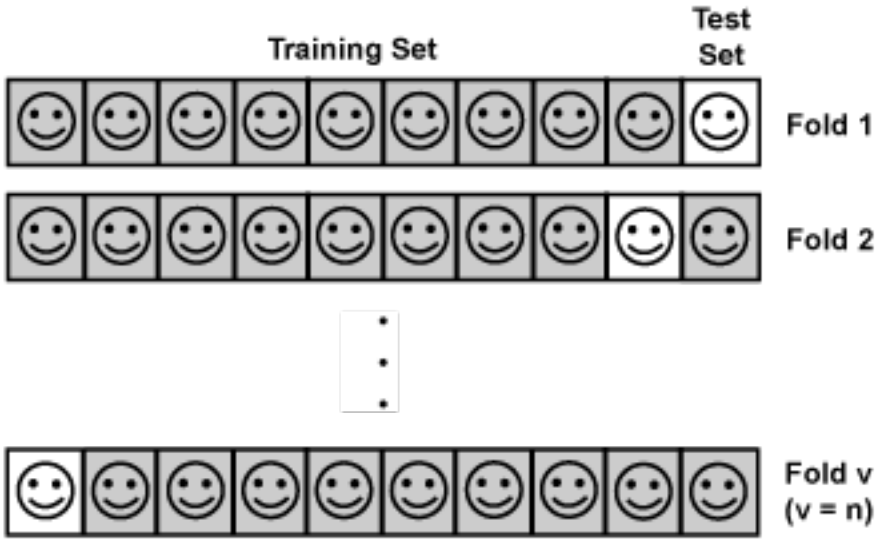

Figure 6: Leave One Out Cross Validation

However, the fact that it is the most computationally intensive does not necessarily make it superior to cross validation with smaller values for v. This is due to the fact that the n test sets are very similar to each other, resulting in a high variance. The computational strain of LOOCV has made it less popular for large data sets, and as a result, its effectiveness in estimating generalization error has not been thoroughly studied (20).

Monte Carlo Cross Validation is a third method of cross validation. It introduces randomness by randomly splitting the cohort into a training set and a test set. This process is repeated many times (e.g., 20, 50, or 1000 times).

It often employs similar splits to v-fold Cross Validation. For example, with each iteration 90% of the observations may be in the training set and 10% in the test set. As the number of iterations increases, this form of cross validation becomes increasingly computationally expensive.

We chose to use (LOOCV) to estimate prediction error of our logistic regression models. Despite its computational burden, LOOCV was chosen in part because we had sufficient computing power to perform it. However, as noted by Molinaro (20), it has a high variance when compared to less extreme forms of v-fold cross validation. Compared to n-fold or Monte Carlo cross validation, this may have created artificially large 95% confidence intervals, decreasing the likelihood that we would achieve statistical significance with our prognostic models when comparing them to the clinical gold standards of TNM and NPI.

Model Creation with Combination Magic

All computations were performed using The R Project for Statistical Computing (RPS), an open-source language and environment. In pre-study trials, the code for creating models was written in R alongside the R code for analyzing the logistic regression model and computing the misclassification.

The limitation of this approach was several-fold. First, while R is an excellent language for statistical analysis, data manipulation, and graphing, it lacks the advanced programming features found in more traditional programming languages. Examples include the availability of classic data structures and strong debugging support. As a result, it was difficult to create

a generalized algorithm for selecting all possible combinations of r from n, and such an algorithm would still not have the flexibility for custom model creation of the form we desired. Second, by integrating model selection into a monolithic analysis run, we would not have the opportunity to selectively re-run part of the analysis in case of failure. Third, a graphical environment for manipulating parameters related to model selection would be significantly more user-friendly and offer improved ability to visualize the results of model construction in advance of our computing runs.

We realized that by extracting the process of model creation from execution of the statistical runs, we could solve each of these issues. A more conventional programming language would provide us with the libraries and GUI environment to create a flexible, generalized tool for model generation. Given our familiarity with Java, this is the language we chose. By generating our models in advance of the analysis run, and writing them out to multiple input files, the failure of any one file would allow the others to proceed, and the failed file could be re-run independently.

We wrote a Java program (Combination Magic) to accomplish this task of model generation. Combination Magic evolved to handle several additional features, which we will describe in the next paragraphs.

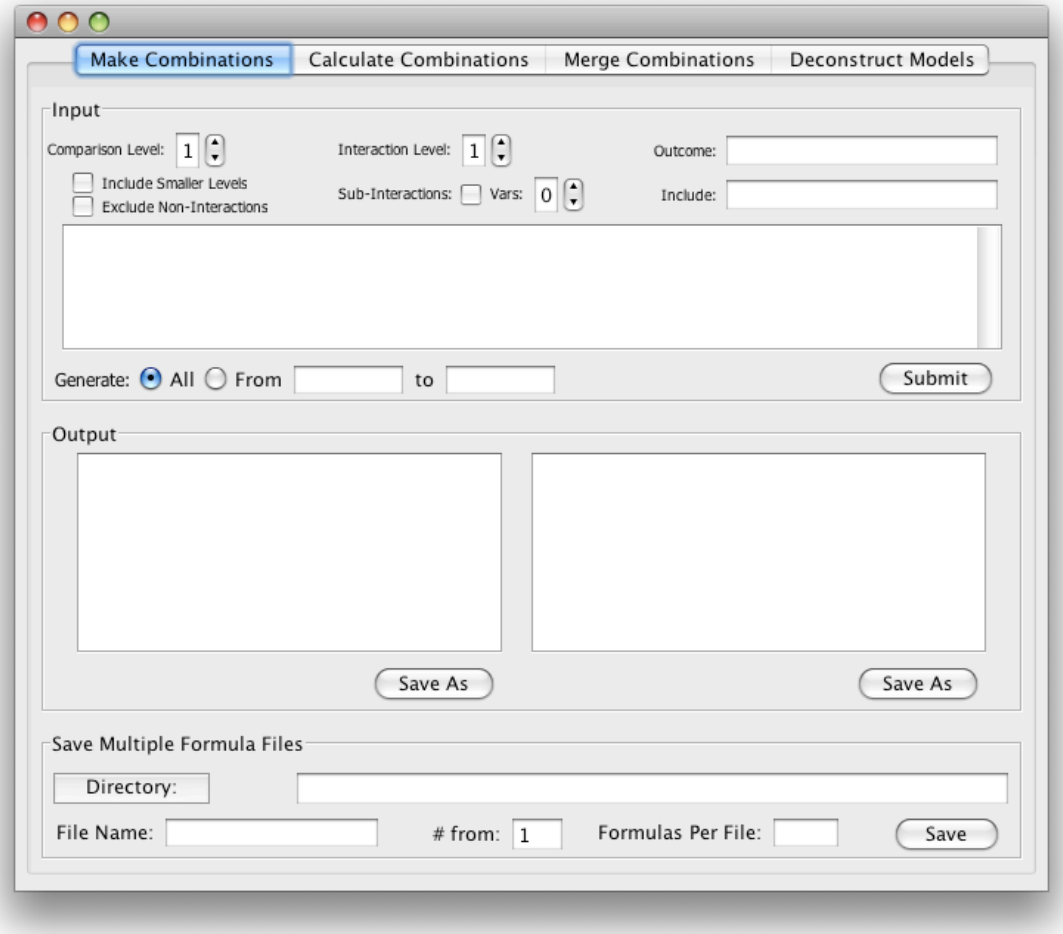

Figure 7: Combination Magic

Most simply, Combination Magic takes a list of variables (of length n) and a "comparison level" (r) parameter that specifies how many variables should be in each combination. For example, asking for all trivariate runs (taking r=3 from n=84) yielded 95,284 combinations (i.e., models) for our data set. (This is according to the formula for combinations of r objects from n choices, n!/[r! * (n-r)!) The program's output is two-fold. The text box on the left contains all the possible combinations of the input variables. In turn, this is the input to the model generator, which creates one or more models for each combination. These models are displayed in the textbox on the right.

Fundamentally, there are two types of models in Combination Magic, which we call “additive” (e.g., A + B + C) and “interactive” (e.g., A*B*C). Note that interactive models, such as “A*B”, should be used when the variables A and B influence each other, with the expectation that the product of these two variables will meaningfully improve the predictive value of the model.

Various parameters control the way that Combination Magic generates models from the combinations. First, an “Include Smaller Levels” checkbox allows the user to include multiple levels in the output; for example, trivariate, bivariate, and univariate. Second, the “Exclude Non-Interactions” checkbox allows you to remove pure additive models from the output. Often you will want to run all additive models, and do all interactive models in a following run. Third, Interaction Level specifies the level at which interactions are produced. Note that this interaction level must be within the range of the Comparison Level (from 2 to Comparison Level if “Include Smaller Levels” is checked, otherwise just Comparison Level) to have any effect. Fourth, the Sub-Interactions checkbox turns on generation of sub-interactions, while the adjacent Vars input designates the maximum number of terms in each sub-interaction. For example, when processing a quadvariate model, Vars = 3 would yield sub-interaction terms that included both two variables and three variables, whereas Vars = 2 would just include the two-variable sub-interactions.

Combination Magic Use Cases

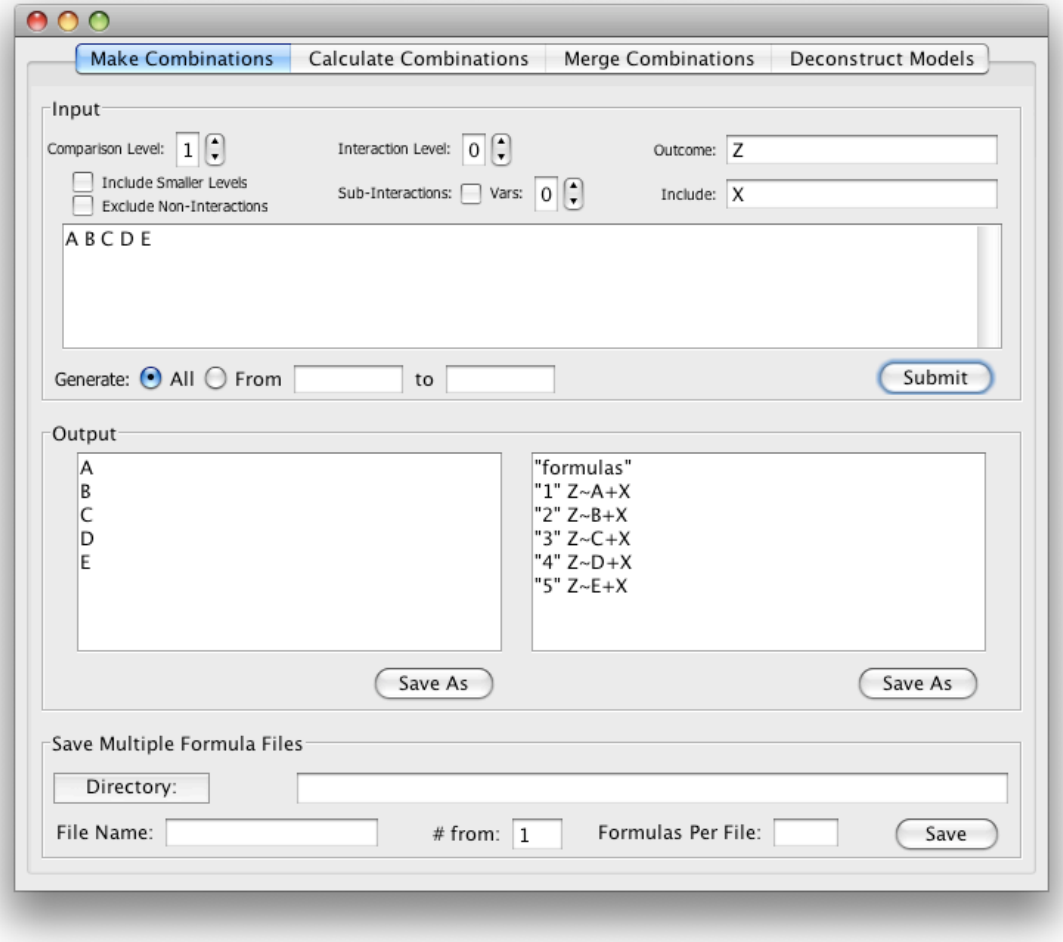

Figure 8: Combination Magic Use Case #1

This is a simple example of model creation. There are five variables and the comparison level is one. Since each variable is only compared to itself, this yields five possible models. Note that we specified Z as the outcome variable. We also specified that the variable X should be included in every formula. Note that any variable included in this manner will not participate in comparisons, although this variable can be of any arbitrary form (e.g., an interaction such as X*Y or a logarithmic variable such as log X).

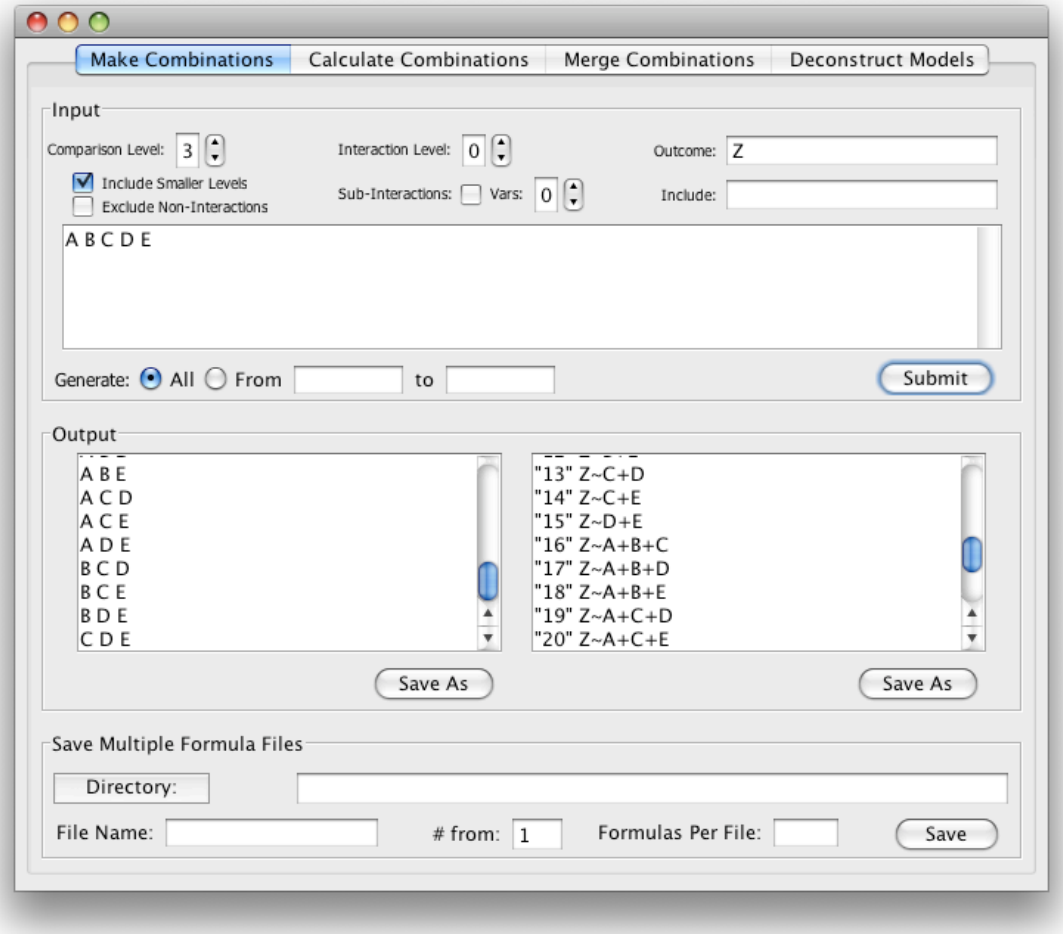

Figure 9: Combination Magic Use Case #2

Now we have specified a comparison level of 3. This creates models of the form "Z~A+B+C". Since we have checked "Include Smaller Levels", we also have models of the form "Z~A+B" (i.e., a comparison level of 2). In total, 25 models are created.

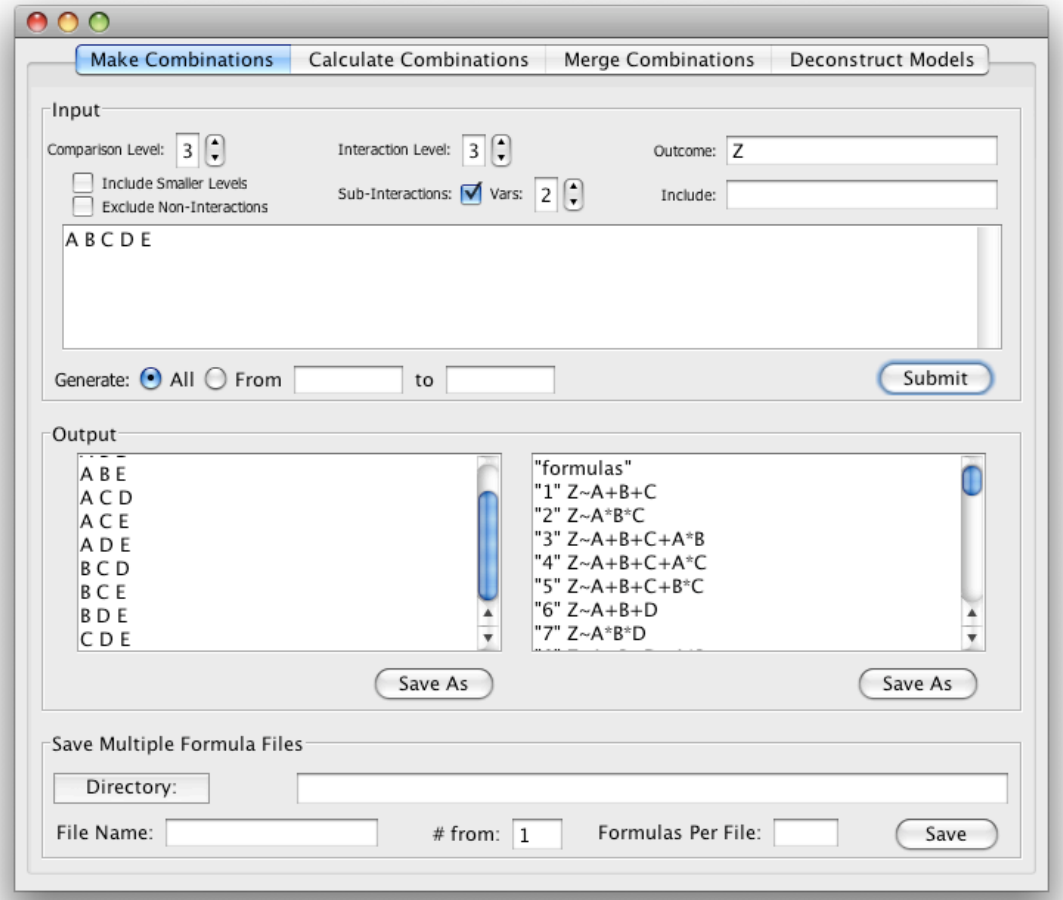

Figure 10: Combination Magic Use Case #3

Now, in addition to a comparison level of 3, we have specified an interaction level of 3. This interaction level creates interactions of the form "A*B*C". Since we have selected the "Sub-Interactions" check box, models will also be created with smaller interactions up to the level specified. In this case, this means the inclusion of interactions of 2 (e.g., A*B). In total, 10 combinations and 50 models are created.

Combinatorial Explosion and Java Heap Restrictions

We quickly realized that combinatorial explosion would not only slow down completion of our runs on the clusters, but also impact our ability to generate the models in Combination Magic due to heap and stack overflows. In order to solve the stack overflow problem, the recursive combination algorithm was rewritten to avoid calling itself. (It is only a pseudo-recursive function in that each call to the function only necessitates one "recursive" call until the base case is eventually reached, not two or more.)

The heap overflow problem was more problematic. Java has a heap size limit on various operating systems. On Windows, it is 1.6 gigabytes. Even after optimization and allocation of the maximum amount of memory, heap size would be exceeded whenever the number of models reached the low millions. To solve this, we added the ability to selectively generate ranges of output (e.g., combinations 500,000 to 999,999, and so forth). We also added a small calculator, available in the "Calculate Combinations" tab, to calculate the number of combinations to expect from a set of parameters r and n. This assisted in our planning of runs, allowing us to generate maximum model output without causing a heap overflow.

Despite these optimizations and workarounds, we reached a limit beyond which we could not search exhaustively. For our data, exhaustively searching the 5-variable space would require examining 30,872,016 models (5 from 84). Because LOOCV requires repeating the generation of each logistic regression model 539 times, alternating leaving out each patient, exhaustively searching

the 5-variable space would actually require more than 16 billion distinct and expensive logistic regression calculations, and that's without interactions, sub-interactions, or custom terms. We did not have enough computer power to accomplish this task. Thus, we needed a new strategy.

For runs beyond quadvariate, instead of selecting from all 100 variables (recall that each PI3-Kinase biomarker has several subcellular compartmental expression levels), we split the variables into two "families". One family consisted of the best subcellular compartmentalizations of the PI3K biomarkers. The second family consisted of the AQUA and pathologist-scored ErbB family markers and ER/PR, along with the clinical variables pTumor, pMet, and pNode. We will refer to these as the PI3K and ErbB families, respectively. 5- and 6-variable runs were performed on each family.

Next, we merged the results from each family into an aggregate run that included the best models from each family. Combination Magic was extended with new functionality to enable the merge. We will use the example of creating a six-variable "merged" run from the best thirty models of the two 5-variable runs of each family. The merge consisted of selecting six variables from the pool of variables created by all pair-wise combinations of the top models from one run with the top models of the second run. Variables were extracted from each pair and then combined into one pool, with redundant variables, when they existed, thrown out. If each 5-variable model lacks redundant variables, this leaves a pool of ten variables, and selecting combinations of six yields 210 combinations.

There are 900 pair-wise combinations of 30 with 30, leading to 900 total pools. Thus, the total number of models is potentially 900*210 = 189,000. In practice, many duplicate models existed and were eliminated, but the elimination of duplicate models was offset by the expansion of sub-interaction terms. Thus, the pool actually involved twelve or more variables. In our case, merging the top thirty models from the PI3K and ErbB families yielded 284,301 unique additive models without sub-interactions or interactions.

After completing a run with additive models only, we went one step farther by also performing logarithmic expansion (each variable's logarithmic term was also added to the pool), treating sub-interaction terms as a variable in their own right, and creating models with interactions and sub-interactions. Due to combinatorial explosion, this required a reduction in the number of models merged. We took the top ten models from the PI3K and ErbB families and merged them together. This yielded 880,000 models with interactions and sub-interactions, not including the additive models that had already been processed.

Summary

In summary, our attempts to find improved prognostic models in invasive breast cancer when compared to the clinical gold standards of TNM staging and the Nottingham Prognostic Index were not statistically significant. The inability to achieve statistical significance was likely multifactorial. Broadly, our focus on biological markers in the PI3 Kinase pathway only may have been insufficient. There are many biological pathways important in human breast

cancer, and their interactions are complex and not fully understood. Subgroup analysis of HER2 positive patients may have increased the significance of our results as the importance of the PI3 Kinase pathway is amplified in this group. Refinement of our statistical methodology may have further increased significance. Imputing missing data points may have lead to more accurate models and narrower confidence intervals. Use of n-fold or Monte Carlo cross validation methods may have also led to narrower confidence intervals. Removing patients older than 80 from our cohort may have reduced the influence of confounding comorbidities. Most significantly, we continue to believe in the potential of biomarker analysis to improve upon existing prognostic models in breast cancer and believe that this is an area deserving of continued attention and research efforts.

References

1. Barr, L.C., and Baum M. 1992. Time to abandon TNM staging of breast cancer. *The Lancet*. 339:915-917.

2. Benson, J.R. 2003. Overview of the TNM system. *The Lancet Oncology*. 4:56-57.

3. Blamey, R.W., Ellis, I.O., Pinder S.E., Lee, A.H.S., Macmillan, R.D. et al. 2007. Survival of invasive breast cancer according to the Nottingham Prognostic Index in cases diagnosed 1990-1999. *Europ J Cancer*. 43:1548-1555.

4. Hudis, C.A. 2007. Trastuzumab – Mechanism of action and use in clinical practice. *N Engl J Med*. 357:39-51.

5. Hollestelle, A., Elstrodt, F., Nagel, J.H.A., Kallemeijn, W.W., and Schutte, M. 2007. Phosphatidylinositol-3-OH kinase or RAS pathway mutations in human breast cancer cell lines. *Mol Cancer Res*. 5;2:195-201.

6. Dillon, R.L., White, D.E., and Muller, W.J. 2007. The phosphatidyl inositol 3-kinase signaling network: implications for human breast cancer. *Oncogene*. 26:1338-1345.

7. RPS Team RDC. *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2005.

8. Dolled-Filhart, M., McCabe, A., Giltnane, J., Cregger, M., and Camp, R.L. et al. 2006. Quantitative in situ analysis of beta-catenin expression in breast cancer shows decreased expression is associated with poor outcome. *Cancer Res*. 66;10:5487-5494.

9. McCabe, A., Dolled-Filhart, M., Camp, R.L., and Rimm, D.L. 2005. Automated quantitative analysis (AQUA) of in situ expression, antibody concentration, and prognosis. *J National Cancer Inst*. 97;24:1808-1815.

10. Breiman, Leo; Friedman, J. H., Olshen, R. A., & Stone, C. J. 1984. *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

11. http://lib.stat.cmu.edu/S/Harrell/help/Hmisc/html/rcorr.cens.html

12. Efron, B. 1979. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*. 7;1:1-26.

13. Breiman, L. 2001. Random forests. *Machine Learning*. 45:5-32.

14. Ishwaran, H., and Kogalur, U.B. 2007. Random Survival Forests for R. *R News*. 7;2:25-31.

15. Thompson, A.M., and Lane, D.P. 2010. P53 transcriptional pathways in breast cancer: the good, the bad and the complex. *J Pathol*. 220:401-403.

16. Ayer, T., Chhatwal, J., Alagoz, O., Kahn Jr, C.E., and Woods, R.W., et al. 2010. Comparison of logistic regression and artificial neural network models in breast cancer risk estimation. *Radiographics*. 30:13-22.

17. Tu, J.V. 1996. Advantages and disadvantages of using artificial neural network versus logistic regression for predicting medical outcomes. *J Clin Epidemiol*. 49;11:1225-1231.

18. Chhatwal, J., Alagoz, O., Lindstrom, M.J., Kahn Jr, C.E., and Shaffer, K.A. et al. 2009. A logistic regression model based on the national mammography database format to aid breast cancer diagnosis. *Amer J Radiol*. 192:1117-1127.

19. Hwa, H., Kuo, W., Chang, L., Wang, M., and Tung, T. et al. 2008. Prediction of breast cancer and lymph node metastatic status with tumour markers using logistic regression models. *J Evaluation in Clinical Practice*. 14:275-280.

20. Molinaro, A.M., and Lostritto, K. 2010. Statistical Resampling Techniques for large biological data analsyis. In *Statistical Bioinformatics: A Guide for Life and Biomedical Science Researchers*. Lee J.K., editor. Hoboken, NJ: John Wiley & Sons, Inc. Ch 10.

R Statistical Code

```
doInit <- function() {

logistic = function(form, dat.sam, samx=samx, res.name, n.fold) {

  options(warn = -1)
  cv = unique(samx)[order(unique(samx))]
  n.fold<-length(cv)
  ret = list()
  error = numeric(n.fold)
  failures = 0
  outcomes = 0

  for( j in cv ) {
    model.input = data.frame(dat.sam[which(samx!=j),])
    names(model.input) = c(names(dat.sam))
    #note: line below returns to original data to get the input row
    outcome.input = data.frame(datx[j,])
    outcome = NA #in case of failure
    fxoutput = tryCatch({
      model = glm(form, family=binomial, model.input)
      outcome = predict( model, outcome.input, type="response")
    }, error = function(ex) {
        failed = "yes"
      } )
    if(!(is.numeric(fxoutput)) && !(is.na(fxoutput))) {
      failures = failures + 1
    } else if(!(is.na(fxoutput))) {
      outcomes = outcomes + 1
    }
    error[j] = as.numeric((outcome > 0.5) != outcome.input$tenyrcens)
  }
  ret[[1]] = sum(error[cv], na.rm=T)/sum(!is.na(error[cv]))
  ret[[2]] = outcomes
  ret[[3]] = failures
  return ( ret )
}

  assign("logistic", logistic, envir=globalenv())

  #read in data from data file
  datx = data.frame(read.table("../YTMA49cleanNPI.txt", sep="\t",
header=T))
  datx$tenyrcens = 1-datx$tenyrcens
  #specify variables as factors here
  datx$TumorType = factor(ifelse(datx$TumorType=="IDC-
NOS",0,ifelse(datx$TumorType=="IDC-
lobft",1,ifelse(datx$TumorType=="IDC-
LowRisk",2,ifelse(datx$TumorType=="ILC",3,NA)))))
  datx$HistoGrade =
factor(ifelse(datx$HistoGrade==1,"Low",ifelse(datx$HistoGrade==2,"Mediu
m",ifelse(datx$HistoGrade==3,"High",NA))))
  datx$NuGrade = factor(datx$NuGrade)
```

```
  datx$Laterality = factor(ifelse(datx$Laterality ==
"L","L",ifelse(datx$Laterality=="R","R",ifelse(datx$Laterality=="B","Z"
,NA))))
  datx$pTumor = factor(datx$pTumor)
  datx$pMet = factor(datx$pMet)
  datx$pNode = factor(datx$pNode)
  datx$PathER = factor(datx$PathER)
  datx$PathPR = factor(datx$PathPR)
  datx$PathHER2 = factor(datx$PathHER2)
  datx$NPI = factor(datx$NPI)
  assign("datx", datx, envir=globalenv())
}

doBootstrap <- function(iter, offset) {
  res.name <- "tenyrcens"
  set.seed(as.integer(iter)) #sets seed for random number generator
  #read in formulas from text file
  formulas =
data.frame(read.table(paste("formulas",iter,".txt",sep="")))
  #only takes one formula per input file
  form = as.formula(as.character(formulas[1,1]))
  n.confidence = 1000
  misclassifications = numeric(1:n.confidence)
  successes = numeric(1:n.confidence)
  failures = numeric(1:n.confidence)
  for( i in 1:n.confidence ) {
    #build an array of the row values to sample
    samx<-sample(c(1:nrow(datx)),nrow(datx),replace=TRUE)
    #the data transformed by sampling
    dat.sam <- datx[samx,]
    #rerun the sampling algorithm if all rows are 0 or 1
    while( sum(dat.sam[,res.name]) == nrow(dat.sam) |
sum(dat.sam[,res.name]) == 0 ) {
      samx = sample(c(1:nrow(datx)), nrow(datx), replace=T)
      dat.sam = datx[samx,]
    }

    ret = logistic(form, dat.sam, samx=samx, res.name=res.name,
n.fold=nrow(dat.sam))
    misclassifications[i] = ret[[1]]
    successes[i] = ret[[2]]
    failures[i] = ret[[3]]
  }

  quants = quantile(misclassifications,c(0.025,0.975),na.rm=TRUE)
  results.names =
list(c(as.character(formulas[1,1])),c("mean","median",".025",".975","su
ccesses","failures"))
  results = matrix(nrow=1, ncol=6, dimnames=results.names)
  results[1,] =
rbind(mean(misclassifications),median(misclassifications),quants[1],qua
nts[2],mean(successes),mean(failures))
  results = round(results,3)
  #write the results out to a numbered text file
  write.table(results,file=paste("results",iter,".txt",sep=""))
}
```

```
#for running the bootstrap
doInit()
doBootstrap(commandArgs()[5], commandArgs()[6])
```

Tables

Table One: AQUA-Measured Variables (Hormonal Receptors and PI3 Kinase pathway)

Variable	Explanation	Variable	Explanation
ER	Estrogen Receptor	PR	Progesterone Receptor
EGFR	Endothelial Growth Factor Receptor	HER2	Human Epidermal Growth Factor Receptor 2
HER3	Human Epidermal Growth Factor Receptor 3	HER4	Human Epidermal Growth Factor Receptor 4
ERK	Extracellular signal-Related Kinases	PTEN	Phosphatase and Tensin Homolog
PI3Kp85	PI3 Kinase Pathway	FOX03	PI3 Kinase Pathway
eIF4E	PI3 Kinase Pathway	p27kip1	PI3 Kinase Pathway
BCL2	PI3 Kinase Pathway	AKT1	PI3 Kinase Pathway
AKT2	PI3 Kinase Pathway	AKT3	PI3 Kinase Pathway
CMYC	PI3 Kinase Pathway	CyclinD1	PI3 Kinase Pathway
FOX01	PI3 Kinase Pathway	MTOR	PI3 Kinase Pathway
NFkB	PI3 Kinase Pathway	p70S6K	PI3 Kinase Pathway
PI3Kp110	PI3 Kinase Pathway		

Table Two: Clinical Variables

Variable	Explanation	Variable	Explanation
DiagAge	Age at Diagnosis	pMet	Metastasis (TMN)
pTumor	Tumor Size (TMN)	pNode	Nodal status (TMN)
HistoGrade	Histologic Grade	NuGrade	Nuclear Grade
Laterality	Light or Right	PathER	Estrogen Receptor (pathologist-scored)
PathPR	Progesterone Receptor (pathologist-scored)	PathHER2	HER2 Receptor (pathologist-scored)
TumorType	Histologic Type		

Complete Univariate Results

Univariate	Mis.Est.	AUC	n	Mean	0.025	0.975
NPI	0.326	0.644	473	0.326	0.292	0.359
~pNodal Stage	0.353	0.597	539	0.372	0.322	0.493

TNM	0.367	0.609		0.367	0.313	0.447
~Metastasis Stage	0.368	0.537	536	0.367	0.334	0.398
~Fox01_NU	0.369	0.524	434	0.375	0.336	0.415
~AKT1_TM	0.373	0.5	415	0.374	0.335	0.412
~PI3Kp110_TM	0.377	0.539	403	0.386	0.343	0.431
~mTOR_TM	0.382	0.5	429	0.383	0.346	0.419
~NFkB_TM	0.383	0.5	439	0.389	0.35	0.433
~HER2_MB	0.385	0.524	535	0.391	0.357	0.425
~PR_NU	0.385	0.5	473	0.385	0.352	0.421
~PTEN_TM	0.386	0.5	446	0.387	0.353	0.434
~p70S6K_NU	0.387	0.506	439	0.389	0.353	0.428
~CyclinD1_TM	0.388	0.5	456	0.389	0.356	0.424
~Nuclear Grade	0.39	0.569	502	0.400	0.358	0.538
~cmyc_TM	0.39	0.506	323	0.393	0.346	0.443
~Tumor Stage	0.392	0.566	502	0.373	0.332	0.506
~p70S6K_TM (repeat)	0.392	0.506	439	0.388	0.351	0.423
~AKT2_TM	0.392	0.5	449	0.395	0.359	0.433
~eIF4E_NN	0.394	0.505	480	0.402	0.364	0.442
~BCL2_TM	0.394	0.5	462	0.396	0.359	0.449
~Age at Diagnosis	0.395	0.5	539	0.399	0.366	0.441
~PI3Kp85_NU	0.397	0.498	466	0.399	0.365	0.436
~Laterality	0.398	0.5	530	0.407	0.367	0.47
~p27kip1_NU	0.398	0.5	427	0.402	0.363	0.439

~ER_NU	0.4	0.5	515	0.423	0.369	0.444
~AKT3_NU	0.401	0.5	362	0.403	0.361	0.446
~FOXO3A_MB	0.401	0.498	401	0.401	0.363	0.442
~PI3Kp85_NN (repeat)	0.401	0.495	466	0.400	0.365	0.435
~AKT3_MB (repeat)	0.403	0.508	362	0.404	0.363	0.449
~p27kip1_MB	0.403	0.5	427	0.404	0.365	0.447
~HER3_NN	0.404	0.511	488	0.402	0.365	0.439
~EGFR_MB	0.405	0.506	514	0.395	0.359	0.432
~PathER	0.407	0.5	509	0.424	0.372	0.642
~HER4_NN	0.409	0.5	472	0.409	0.373	0.446
~PathPR	0.409	0.5	494	0.433	0.377	0.664
~AKT3_NN (repeat)	0.412	0.497	362	0.403	0.362	0.444
~ERK_TM	0.413	0.5	404	0.417	0.377	0.466
~TumorType	0.423	0.515	539	0.406	0.365	0.462
~Histologic Grade	0.460	0.510	265	0.500	0.414	0.766
~PathHER2	0.471	0.5	499	0.423	0.379	0.492

Selected Multivariate Results

5-variate	Mis. Est.	AUC	n	Mean	0.025	0.975
Fox01_NU + AKT1_NU + mTOR_MB + p70S6K_NU + AVG_BCL2_TM + Fox01_NU*AKT1_NU*mTOR_MB	0.295	0.587	285	0.33	0.274	0.393
Fox01_NU + AKT1_NU + mTOR_MB + AVG_BCL2_TM +	0.302	0.574	285	0.33	0.277	0.391

p70S6K_NN + Fox01_NU*mTOR_MB*p70S6K_NN						
Fox01_NU + PI3Kp110_TM + mTOR_MB + p70S6K_NN + FOXO3A_NN + Fox01_NU*PI3Kp110_TM*mTOR_MB*FOXO3A_NN	0.301	0.666	279	0.337	0.28	0.401
ER_NU + HER3_NN + HER4_NN + pMet + pTumor + HER3_NN*pTumor	0.310	0.663	393	0.345	0.294	0.399
ER_NU + HER3_NN + HER4_TM + pMet + pTumor + HER3_NN*pTumor	0.315	0.659	394	0.348	0.295	0.406
HER3_NN + log(HER4_NU/HER4_NN) + PathPR + PathHER2 + pTumor + log(HER4_NU/HER4_NN) *PathPR*PathHER2	0.316	0.702	380	0.355	0.301	0.411
log(HER4_NU/HER4_NN) + PathER + PathPR + pMet + pNode + PathER*PathPR*pNode	0.317	0.699	441	0.373	0.314	0.443

6-variate	Mis. Est.	AUC	n	Mean	0.025	0.975
Fox01_NU + AKT1_NU + mTOR_MB + AKT2_NN + AVG_BCL2_TM + p70S6K_NN + Fox01_NU*AKT1_NU*mTOR_MB	0.295	0.593	285	0.331	0.273	0.401
Fox01_NU + AKT1_NU + mTOR_MB + cmyc_NU + AVG_BCL2_TM + p70S6K_NN + AKT1_NU*p70S6K_NN	0.297	0.627	195	0.333	0.266	0.403
Fox01_NU + AKT1_NN	0.295	0.590	285	0.334	0.277	0.403

+ mTOR_MB + p70S6K_NU + AKT2_NN + AVG_BCL2_TM + Fox01_NU*mTOR_MB*p 70S6K_NU						
ER_NU + HER3_NN + HER4_NN + log(HER4_NU/HER4_N N) + pMet + pTumor + HER3_NN*pTumor	0.313	0.663	393	0.348	0.298	0.41
ER_NU + HER2_MB + HER3_NN + HER4_MB + HER4_NN + pTumor + HER3_NN*pTumor	0.311	0.658	395	0.348	0.3	0.402

6-variate "Merge"	Mis. Est.	AUC	n	Mean	0.025	0.975
AVG_BCL2_TM + Fox01_NU*AKT1_NU*mTOR_MB + HER4_MB + HER4_NN + p70S6K_NU + pTumor + HER4_NN*pTumor	0.247	0.728	243	0.303	0.240	0.370
AKT1_NU + AVG_BCL2_TM + Fox01_NU*mTOR_MB*p70S6K_NN + HER4_MB + HER4_NN + pTumor + HER4_MB*pTumor	0.259	0.704	243	0.304	0.243	0.379
AKT1_NU + AVG_BCL2_TM + Fox01_NU*mTOR_MB*p70S6K_NN + HER4_MB + HER4_NN + pTumor + HER4_NN*pTumor	0.259	0.713	243	0.304	0.238	0.371
AVG_BCL2_TM + Fox01_NU*AKT1_NU*mTOR_MB + HER4_MB + HER4_NN + p70S6K_NN + pTumor + HER4_NN*pTumor	0.247	0.728	243	0.305	0.241	0.374
AVG_BCL2_TM + Fox01_NU*AKT1_NU*mTOR_MB + HER4_MB + HER4_NN + p70S6K_NN + pTumor + HER4_MB*pTumor	0.259	0.731	243	0.305	0.240	0.378
Fox01_NU + Fox01_NU*AKT1_NU*mTOR_MB + HER3_NN*pTumor + HER4_NN + p70S6K_NU + pMet + Fox01_NU*HER3_NN*pTumor	0.258	0.690	260	0.325	0.263	0.394