Predictive Modeling with JMP Genomics

Introduction

Predictive modeling or biomarker discovery with modern genomics data is a popular activity and is a critical component in delivering on the promise of the underlying technologies. In addition to producing predictions of important binary, continuous, or time-to-event traits, the methods can provide understanding of the scientific mechanisms at play. Ideally, a small set of predictors can be gleaned from a much larger set comprised of genotypes, expressed genes, small molecules, and/or other clinical indicators and then used to predict endpoints like disease state with a high degree of accuracy and in an interpretable fashion. In practice, there are many difficulties and pitfalls that must be overcome to ensure that biomarkers from genomic studies are selected in an unbiased manner. Reproduction of published biomarker signatures has often been problematic, likely as a result of inconsistencies, overfitting, and bias both in modeling methodology and the software used to implement it.

The MicroArray Quality Control (MAQC) consortium recently published recommendations for best practices for predictive modeling with expression data. In a series of papers that appeared in *Nature Biotechnology* and *The Pharmacogenomics Journal*, MAQC authors compared results from numerous data analysis teams that used diverse software and approaches to assess the same six data sets. The consortium observed significant variation in performance across the various endpoints considered, the predictive models used, and the teams deploying them.

While no predictive modeling software can comprehensively and completely automate your biomarker discovery projects, the processes available in JMP Genomics allow you to select from a wide variety of methods, clearly define the parameters used in each model, and compare them rigorously via several common performance metrics. In this document, we will demonstrate the application of a number of these predictive modeling tools in a step-by-step case study. We will use the GSE20194 breast cancer data sets which are featured in MAQC Phase II publications and can be obtained from Gene Expression Omnibus.

Data Set Characteristics

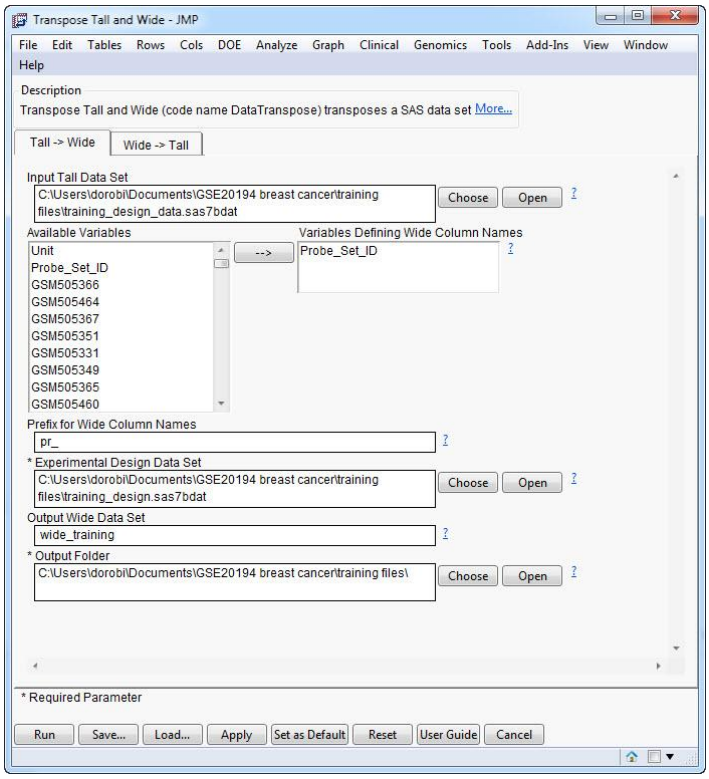
GSE20194 consists of two independent Affymetrix HG-U133A data sets, one with 130 samples and the second with 100 samples. Each set of Affymetrix CEL files is imported separately into a SAS data set using standard RMA parameters. (See the **Data Import** document for details importing CEL files.) No subsequent filtering or normalization steps are performed.

Data Set Preparation

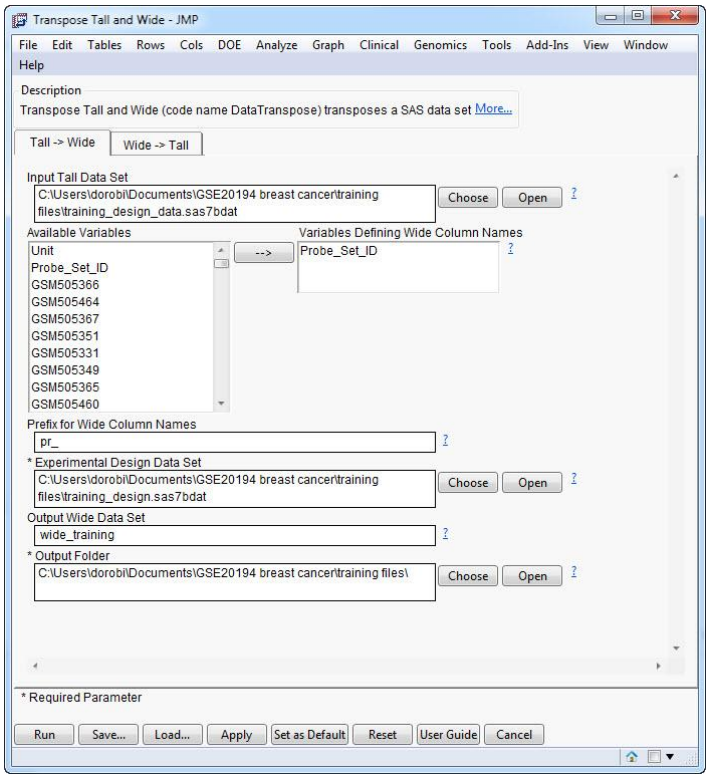
The imported data consists of a tall data set with probe identifiers and intensities in rows and samples in columns, and a separate design data set with sample information. While the tall form is convenient for quality control, normalization, and row-by-row modeling, a wide form is preferred for predictive modeling. To go from tall to wide forms, the tall data is transposed into a wide data set and joined with the sample information in the design data set. Additionally, when working with separate training and test sets, the dependent variable (endpoint) and expression predictors must have the same variable names in both data sets. For categorical dependent variables, category levels must also be named consistently across the two data sets.

Before combining intensity and design information, it is recommended that you have a relatively short length for the name of the dependent variable in the design table. There are length constraints in the software that require dependent variable names to be less than 15 characters. If your dependent variable name is longer than this, please shorten it before proceeding.

1. Select **Predictive Modeling > Predictive Modeling Utilities > Transpose Tall and Wide** from the **Genomics Starter**.

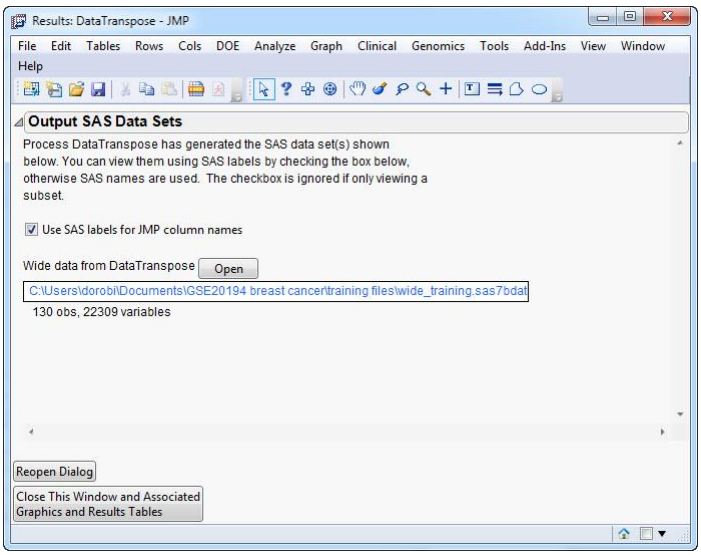


- Fill out the **Transpose Tall and Wide** Analytical Process (AP) dialog as shown below (note path names will be specific to your computer and not exactly as shown below).



- The **Variables Defining Wide Column Names** field should contain the probe set identifier. If the identifier is not unique for each row in the tall data set, a second identifier (e.g., row number) must be included to create unique column names.
- The **Prefix for Wide Column Names** field should contain a character prefix. In this case we are using pr_ to denote probe. The prefix placed in this field should have as few characters as possible and not be the same as the beginning of any other variable name in the design table. As you will see, including this prefix provides the option to use list-style specification fields as a shorthand way to identify groups of predictors.

2. Click **Run** to start the process and wait until a SAS Message window displays the output file name and file path, as shown below.



The number of observations is the number of rows in the newly created wide data set; this equals the number of samples/individuals in the data. The number of variables is the number of columns, which are comprised of probe-level gene expression data plus all of the design file columns. Here we have over 22,000 gene expression measurements to predict only 130 outcomes. Overfitting the data is a very serious risk and care must be taken at each analysis step to avoid it.

Predictive modeling in JMP Genomics is not limited just to one kind of genomic data, such as gene expression measurements. If other data, such as genotypes, are available on the same individuals, you can include them as additional predictors. You can join wide tables by using the **Data Set Utilities > Tables > Merge** function. Refer to the JMP Genomics User Guide for details on the use of the Merge process.

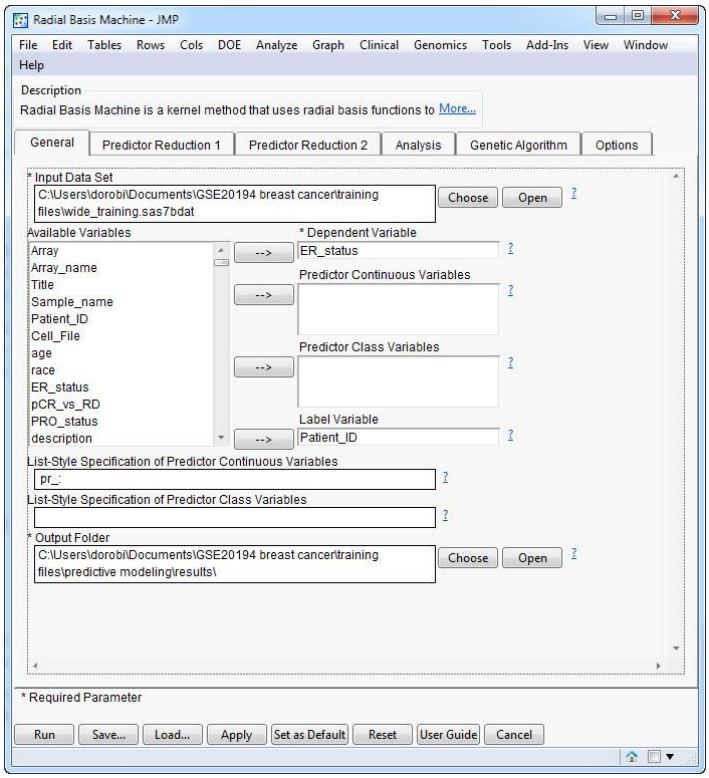
Cross Validation Set-up

There are two primary steps in setting up a cross validation predictive model comparison. The first is creating settings for the models to be cross validated and second is selecting the hold out method.

- Before proceeding, create a new folder in which to store the settings you will create. When running predictive modeling in JMP Genomics, you may run the same processes multiple times, creating new files and overwriting older copies. Creating and using a defined settings folder ensures that you always have the original settings used in all runs of all processes. Also, create a new output folder for the predictive modeling results.

There are a number of predictive modeling algorithms available in JMP Genomics and each offers a variety of different options. We will cover some commonly used methods and options in this document, but this is not meant to be a comprehensive overview of all processes. Refer to the Predictive Modeling chapter of the JMP Genomics User Guide for more information.

1. From the **Genomics Start Menu,** select **Predictive Modeling > Main Methods > Radial Basis Machine**.



- **Choose** the transposed wide data set for the **Input Data Set**. If there are more than 5,000 columns in the data set, you will see a JMP Genomics message listing the number of variables and suggesting that you will want to use list-style specification when referring to the variables.
- The names of the first 5,000 columns in the input SAS data set are displayed in the **Available Variables** field. Note SAS data sets may also have labels for each column that can be different from the names.

*Note: For all predictive modeling APs, the **General, Predictor Reduction 1,** and **Predictor Reduction 2** tabs are identical for all models with the exception of the **Survival** AP. Each model will have **Analysis** options that are specific for that model and take advantage of the model's capabilities and parameters.*

- In this instance, we are using **ER_status** as the dependent variable. This variable is binary and indicates whether or not each breast cancer sample was Estrogen Receptor positive or negative.

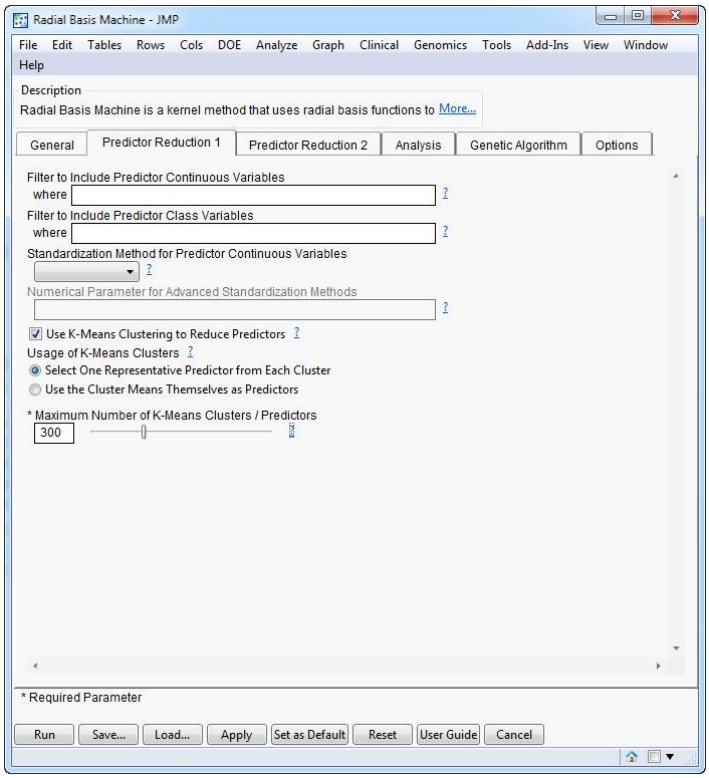
Predictor variables are of two general types: continuous (quantitative) and class (qualitative). The former are numeric and are used directly. The latter are typically nominal (e.g. character-coded genotypes), and are converted to sets of 0-1 variables indicating the nominal level. One method for specifying continuous and class variables is to select variables from the **Available Variables** window, and then use the arrow buttons to move them over to the appropriate field. Here we leave these fields blank because there are too many variables to select directly.

- Patient_ID is specified as the **Label Variable** in order to label points in subsequent output.
- Next are two fields for **List-Style Specification** corresponding to continuous and class predictor variables. The term pr_: is entered into the continuous variable field—this captures all of the gene expression measurements with a simple syntax. The colon at the end of the string indicates that all variables that begin with pr_ (identical to the **Prefix for Wide Column Names** specified during the transpose step) will be entered into the predictive model.
- **Choose** the **Output Folder** as the new folder you created above and then click on the **Predictor Reduction 1** tab.

When performing predictive modeling, it is important to consider the impact of overfitting a model. That is, with many predictors and few samples, a seemingly good model can be created when in fact the model is highly specific to that data set and yet performs poorly when tested with other data sets. A general rule is that you should not have more predictors than samples entering into the model, although certain methods like Radial Basis Machine and Partial Least Squares do accommodate more.

JMP Genomics performs predictor reduction (also known as feature selection) as part of the cross validation process. That is, within each iteration, predictor reduction is performed anew on the training set and then those predictors are applied to the holdout test set. As a result, we gain an additional level of predictor validation, as typically a different set of predictors is chosen for each distinct training set. If the predictors remaining after an iteration of predictor reduction are dependent on which set of samples gets included in the test data set, then they are not considered to be robust predictors in general. Pre-selecting predictors, such as choosing a subset of statistically significant genes as determined by other analysis methods (e.g., ANOVA), will introduce bias into the modeling process. Consequently, we highly recommend that the initial data set contain all potential predictors.

2. Click the **Predictor Reduction 1** tab.
    - The **Predictor Reduction 1** tab contains initial filters and a *K*-means clustering option.



*Note: Criteria specified on the Predictor reduction tabs are applied in order. If options are specified on the **Predictor Reduction 1** tab, then only predictors that pass this filter are used for **Predictor Reduction 2**. You may specify filters on either or both of these tabs.*

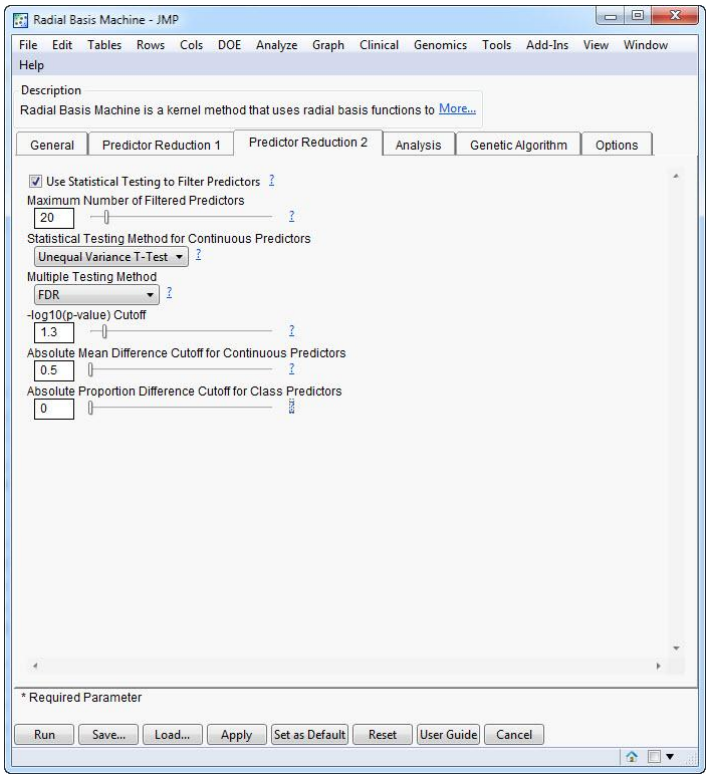
The **Filter to Include…** fields are used for restricting sets of predictor variables. For example, if you have both genotypes and gene expression measurements as predictors, you may wish to only consider one set of these for a particular model fit.

- For this example, we leave these fields blank.

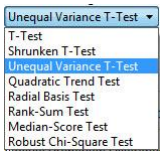
Continuous variables can be standardized in this step by selecting a standardization method in the dialog. Standardizing is useful when predictors are on different scales. When you select this option, SAS PROC STDIZE is called behind the scenes to perform this step. Further predictor reduction is possible using *k*-means clustering, which groups multiple predictors. Depending on the option chosen, *k*-means will either use the predictor closest to the cluster mean or the cluster mean itself as a predictor for the group. The maximum number of clusters constrains the maximum number of predictors

that can be passed into the next step, but it is possible that a smaller number of predictors may be selected. We perform *k*-means clustering here with a maximum of 300 clusters and select one representative from each cluster. This is a great way to remove redundancies in a large predictor set.

3. Select the **Predictor Reduction 2** tab.



This tab sets the criteria for applying statistical tests to the predictors, which are applied now to the variables selected from *k*-means clustering. In this case, the maximum number of predictors will be 20, and we use an unequal variance *t*-test to filter them. Several statistical methods are available at this step, with options for continuous predictors shown below. Class predictors are filtered with the Fisher Exact test, and if the dependent variable is continuous, it is discretized into seven groups prior to testing. If the dependent variable is a class variable with more than two levels, all combinations are tested.

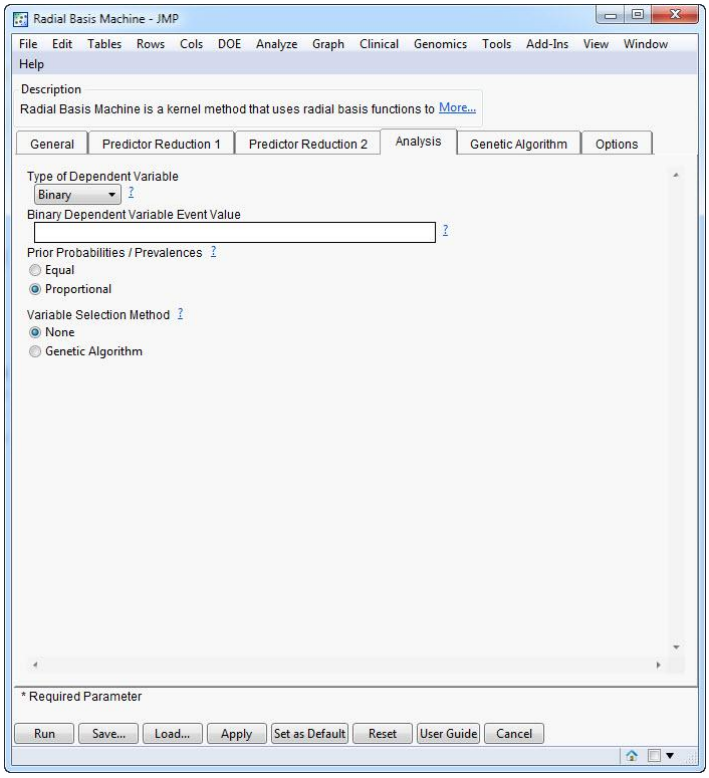


In addition to choosing a specific number of predictors, you can utilize multiple testing corrections by specifiying a **–log10(p-value) Cutoff**. A $-log_{10}$(*p*-value) of 1.3

corresponds to p=0.05. The **Absolute Mean Difference Cutoff** or **Absolute Proportion Difference Cutoff** can be set for Continuous or Class predictors, respectively.

- Set a mean difference value is set at 0.5. As the expression data are log_2 normalized, this corresponds to a minimum of a 1.4-fold mean difference (between the two levels of ER_status) that must be achieved for the predictor to enter the model. Setting a fold-change cutoff in this fashion will often produce a more reproducible set of predictors.

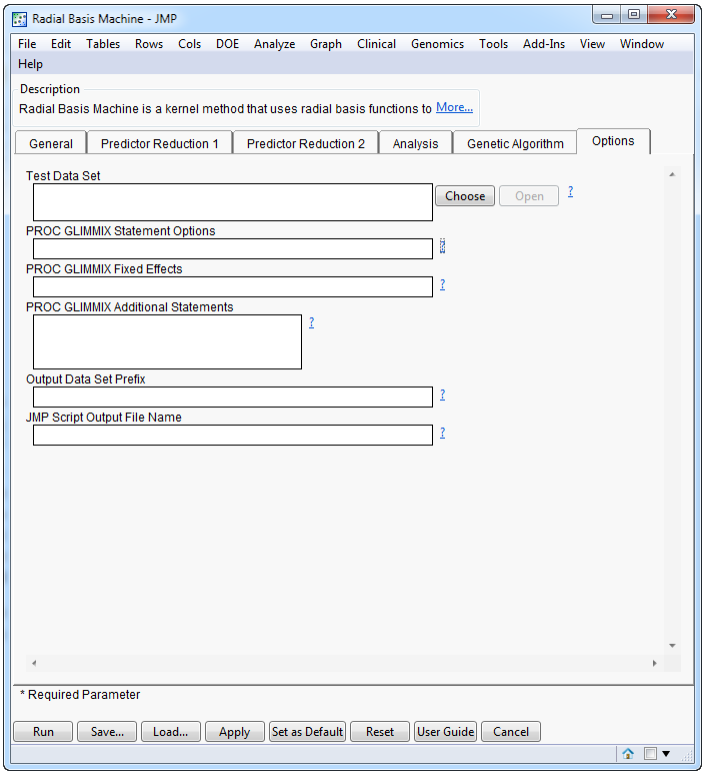
4. Select the **Analysis** tab.



- The options on the **Analysis** tab are different for the different modeling APs and reflect the options that are available for each algorithm. In the Radial Basis Machine dialog, the **Type of Dependent Variable** is specified as being either Continuous or Binary, and here we choose Binary (reflecting the nature of ER_status). In addition, you can specify the **Binary Dependent Variable Event Value**. If this is blank then the second ordered category is chosen by default. You can choose the type of **Prior Probabilities/Prevalences** used to calculate the posterior probability estimates. Here we select Proportional, which uses the observed proportions of ER_status from the input data set as the prior probabilities.

- You can optionally employ a **Genetic Algorithm** to select variables. We do not select this here given previous choices for filtering. In addition, this method can be very computationally intensive.

5. Select the **Options** tab.

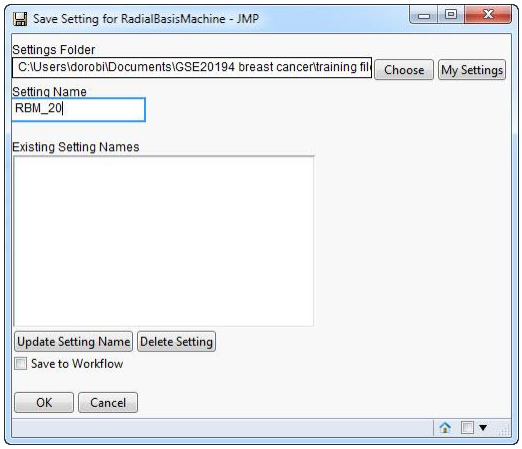


- On all predictive modeling APs, you can specify an external test data set file here when running a whole model fit for training vs. a specific test set. We leave this blank for now. This tab can contain other AP-specific options available. For this example, Radial Basis Machine offers some more advanced general options relating to SAS/STAT PROC GLIMMIX. We leave these blank.

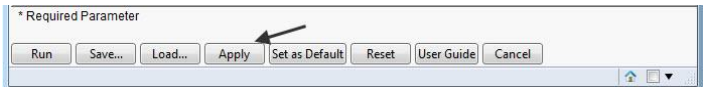
Saving the Setting

- Our model specification is now complete and we now save it for future use. At the bottom of the AP window, select the **Save** button

- Using the **Choose** button, navigate to the settings folder you created earlier, or copy-and-paste the folder directly into the text field.



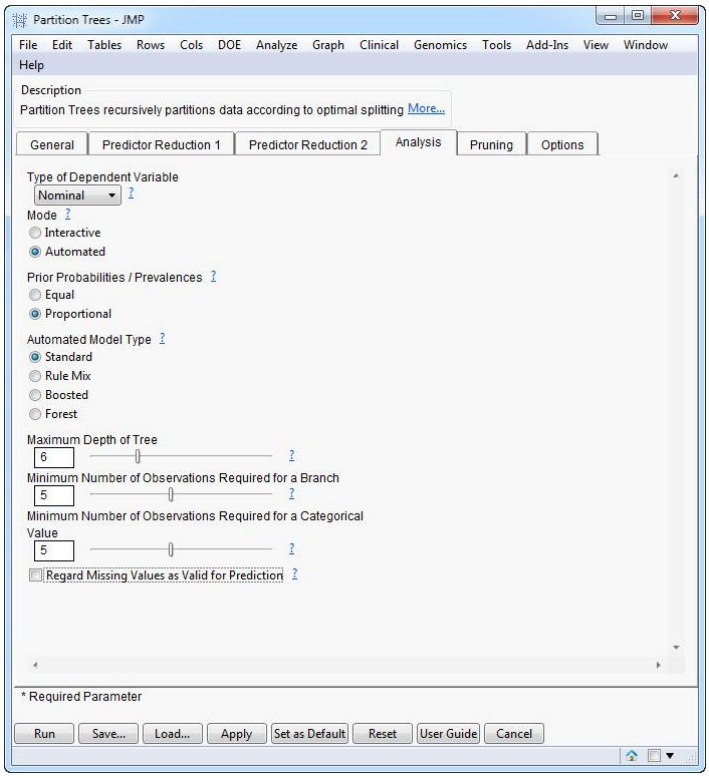
- Type in a setting name that has some description of what is in the model. Here we use “20” in the name to reflect the fact that the number of predictors will be filtered down to 20. Use an underscore (“_”) to separate words or numbers. When finished, click **OK**, and the setting is saved.
- Click the **Apply** button at the bottom of the AP window.



This instructs JMP Genomics to remember the parameters you have specified and apply them in all subsequently opened dialogs for the remainder of the session. Any AP that is opened will contain the same values for identical fields.

- To turn off this temporary setting, select **General Utilities > Clear Parameter Defaults** from the **Genomics Starter** menu.
- To set up a second model, from the **Genomics Starter**, select **Predictive Modeling > Main Methods > Partition Trees**. Examine the values in the **General, Predictor Reduction 1** and **Predictor Reduction 2** tabs and confirm that they are filled out identically to those in the Radial Basis Machine.

6. Select the **Analysis** tab.



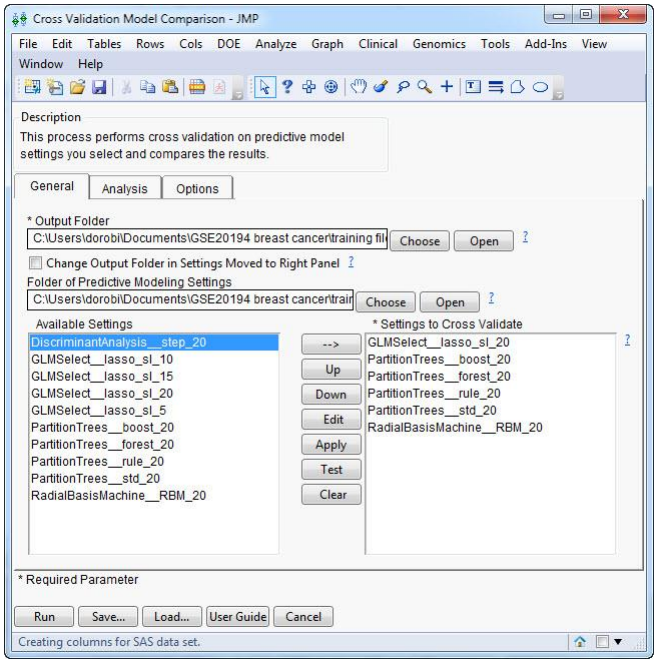
For running cross-validation, you must always select the **Automated** mode whenever there is a **Mode** option. In the Partition Trees AP, you can select from four general kinds of tree-based algorithms, and for each tree-type you can set the depth and minimum numbers of observations for each branch and categorical values. The **Pruning** tab contains further options for the **Rule Mix, Boosted,** and **Random Forest** models.

To save this model, click the **Save** button at the bottom of the AP, and save it in the same directory as before. Continue creating an arbitrary number of desired models and saving them. JMP Genomics is very flexible and powerful in this respect. In the MAQC project, it was used to compare 100s models simultaneously. Keep in mind that execution time will increase roughly linearly with each model you specify.

Cross Validation Model Comparison (CVMC)

You can cross-validate and compare saved model settings to find which models have the best prediction performance. One iteration of a cross validation sequence consists of creating a hold-out test data set, performing predictor reduction and model fitting on the training data set, then applying predictions of that model to the hold-out test data set. Since predictor reduction is performed within each cross-validation iteration, a distribution of selected predictors can be examined as a part of the output.

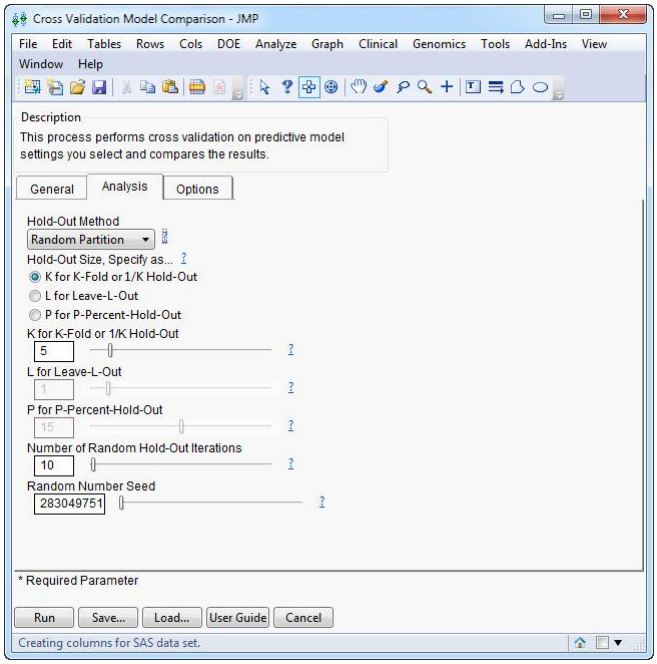
1. Select **Predictive Modeling > Model Comparison > Cross Validation Model Comparison** from the **Genomics Starter**.



- **Choose** the output folder. To keep from overwriting results, make sure this directory is unique for every analysis that you want to retain for later review.
- **Choose** the folder that contains the model settings that were created earlier.
- Highlight the desired settings in the **Available Settings** in the left window, and use the **Arrow** to move them to the right window as **Settings to Cross Validate**. On the right hand side, highlight and remove any model settings you do not want to compare.

Note: All settings on the right hand side must have the same input data set and dependent variable.

2. Click on the **Analysis Tab**.



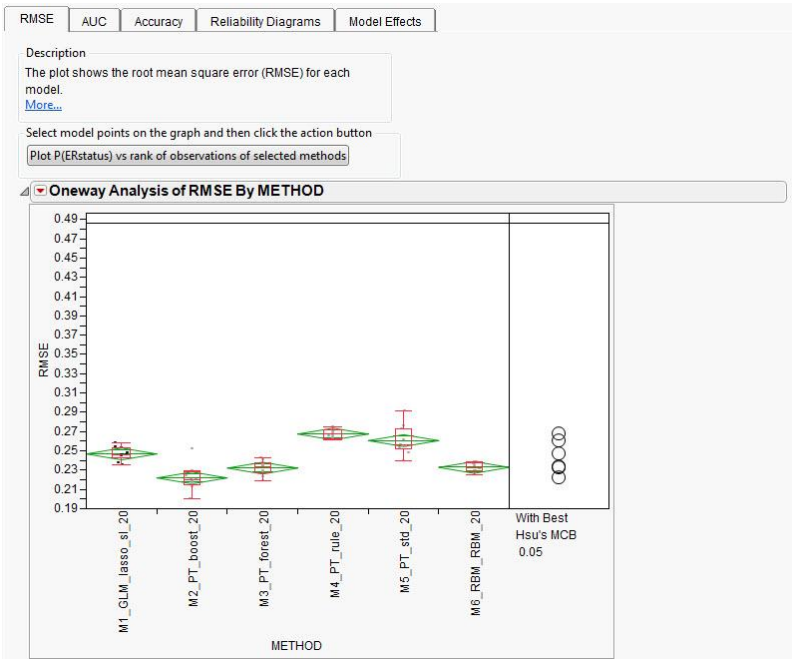
- Select the **Hold-Out Method** from the pull-down menu.
    - **Simple Random** holds out a fraction of the data selected completely randomly.
    - **Stratified Random** holds out a fraction of the data selected randomly within each trait group.
    - **Random Partition** randomly partitions all of the data into groups and holds out each group in turn.
    - **Block Partition** partitions all of the data into blocks of consecutive observations and holds out each group in turn.
    - **Split Partition** partitions all of the data into k parts, with the ith part consisting of observations i, i + k, i + 2k, ...
    - **All Subsets** holds out all possible subsets of a certain size. Currently only subsets of size 1, 2, or 3 are available.
- Next, select the method for defining the **Hold-Out Size**. Define the size in the active window that corresponds to the hold-out size specification.
- Define the **Number of Random Hold-Out Iterations**. The first time you run any new cross validation set, it is recommended that you set the number to 2 to test the cross-validation process as at times errors will not become evident until the end of the cross validation process.

Keep in mind that the run-time for each model can vary and the total time within each model is linear with respect to the number of iterations. In the above example, there are 6 models and 10 iterations each, resulting in 60 iterations total.

3. In the **Options** tab, you can set the number of samples for a test run if desired (decreasing the number of rows in the input data set), and if you are adding to an existing cross-validation result, you can set the start and stop number for the iterations.
4. Click **Run**.

CVMC Results

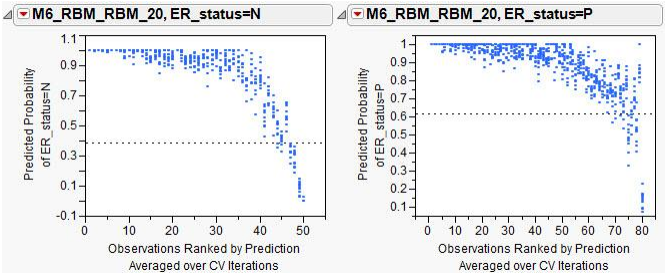
The results from CVMC provide information regarding the best model fit and also the ranked predictors from each model.



The first tab shows a one-way ANOVA on the Root Mean Square Error (RMSE) for each model. This is computed by taking the square root of the average square deviation of each predicted probability from the true value of 0 or 1 (corresponding to the levels of ER_status), so smaller values indicate better performance. A distinct RMSE value is plotted for each iteration and box plots and mean diamonds are overlaid as summary graphics. The horizontal reference line just below 0.49 indicates the expected value from a simple model based on the observed prevalence.

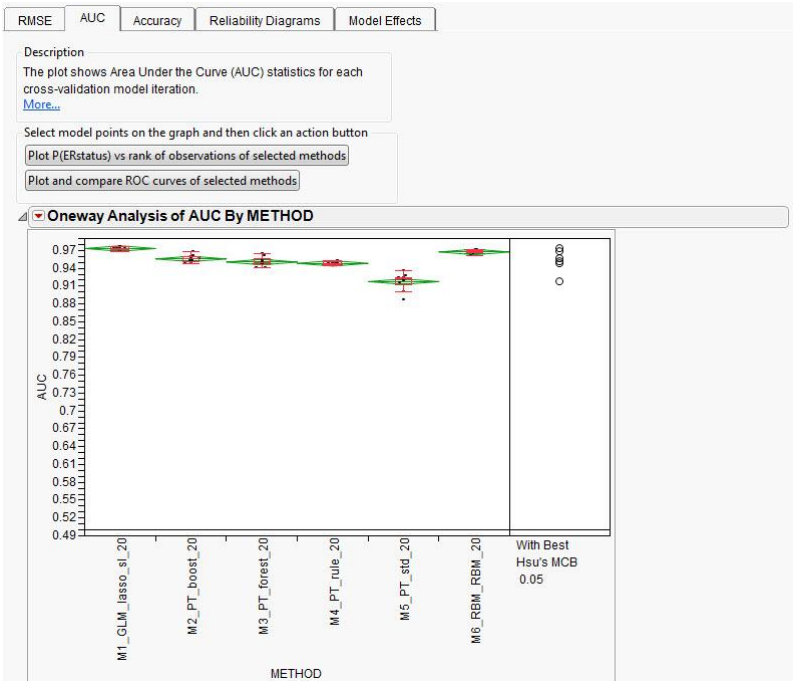
Tabular reports below the figure provide specific values for each models mean, standard deviation, etc. Ideally, you would like to see low RMSE mean values with a tight distribution about the mean. This would indicate that the model fits well and that the fit does not display a high variance across the data sets selected as the training and test sets during the cross validation.

- Using the mouse, select all points and click on **Plot P vs. Rank of Observations** above the figure.



The resulting plots show the summary of accuracy of each sample for each iteration in the cross validation process. The *y*-axis is the predicted probability of each response, and the *x*-axis is the observation numbers ranked from high to low across the iterations. Ideally, there will be many points above the dotted line which indicates the proportion of samples with that trait value in the data set. Observations below the line are those which were predicted poorly when included in test sets. You can view and explore the JMP table behind this plot by double clicking on the table named pred_prob in the **JMP Window List** window.

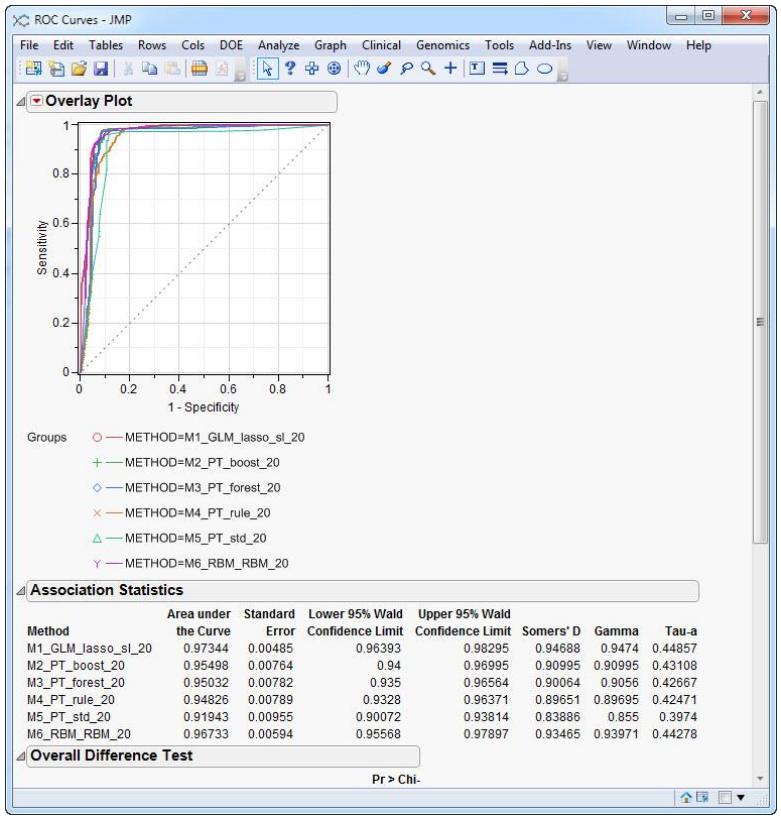
1. Click on the **AUC** tab.



This result shows a one-way ANOVA on the Area Under Curve (AUC) statistics from Receiver Operating Characteristic (ROC) plots for each model. Note that this display is created only when the dependent variable is binary; that is, it is categorical with two

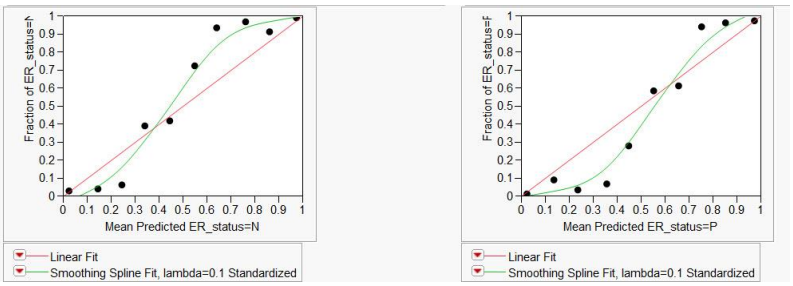
levels. The *y*-axis is the AUC and the x-axis is each model and the results are from a one-way plot as described in the RMSE tab, although here larger is better.

- Using the mouse, select at least one point from each model.
- Click on the **Plot and Compare ROC curves** button above the graph.



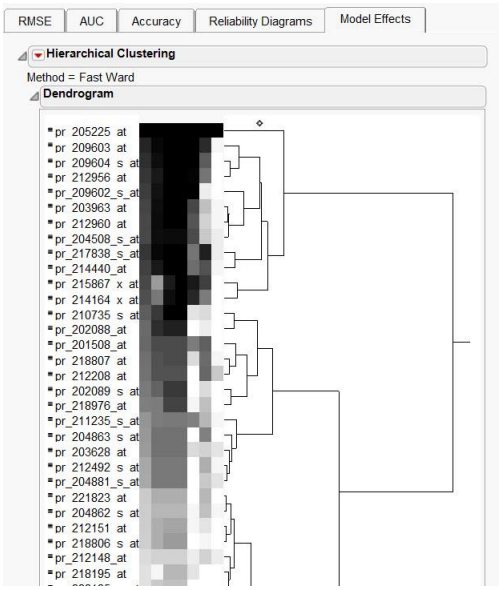
An average ROC plot for each of the models is created, and associated statistics are shown below the graph. Note that in this case, the General Linear (GLM) model has the highest area under curve. There is also an overall test of the null hypothesis that all of the models have the same AUC.

2. Click on the **Accuracy** tab. The format of these results is identical to the previous two tabs and shows the percentage of correct calls for each iteration of each model.
3. Click on the **Reliability Diagrams** tab.



The two reliability diagrams show the fraction of each level of the dependent variable, in this case ER (estrogen receptor) status, vs. the mean predicted value for each level. The closer the points are to the diagonal red line, the better the model fit. In the above figure, there are slight deviations at the high and low end for each indicator, but the overall fit is good.

4. Click on the **Model Effects** tab.



This dendrogram shows the effect usage of each predictor overall and their usage in each model ranked from high to low overall. To be included in these results, each predictor must have been included in the model for at least one cross-validation iteration. In this case, pr_205225_at is the most prevalent predictor and was chosen by all of the models. This is evidence that this predictor is a very good one.

5. In the **Tabs** menu in the upper left-hand side of the results window, click on **Model Effects** and select **View Data**.

This data table can be sorted from high to low on any column. (Note that if you sort it a new data table will be created as the original table's order is linked to the figure.) A section of the table is shown below.

	name	OVERALL	M1_GLM_lasso_sl_20 Model Count	M1_GLM_lasso_sl_20 Model Count/50	M2_PT_boost_20 Model Count	M2_PT_boost_20 Model Count/50	M2_PT_boost_20 Average Importance
1	pr_205225_at	1	50	1	50	1	1
2	pr_217838_s_at	0.693333	41	0.82	50	1	0.198958
3	pr_209603_at	0.79	38	0.76	50	1	0.18427
4	pr_215867_x_at	0.593333	32	0.64	42	0.84	0.049013
5	pr_214440_at	0.633333	27	0.54	50	1	0.119107
6	pr_209604_s_at	0.733333	24	0.48	50	1	0.21615
7	pr_201508_at	0.416667	23	0.46	29	0.58	0.067814

Probe pr_205225_at is the top predictor overall. The column **M1_GLM…Model Count** contains the number of times the predictor entered the model. Pr_205225_at was in the model for all 50 iterations in **M1…** and pr_217838_s_at was in **M1…** for 82% of the iterations. Turning our attention to **M2_PT_boost…** we can see that pr_205225_at, pr_327838_s_at and pr_209603_at and all other probes with a **Model Count** of 50 entered into the boosted model 100% of the time. The **M2_PT_boost_20 Average Importance** shows how important the predictor was in the tree structure. The lower the value, the further down in the splits. So we can see that pr_205225_at was consistently used as the primary differentiator each each iteration of the boosted trees model, and the other predictors were consistently lower.

The combination of these results allows you to find the best model and the highest ranking predictors for the models. You can further refine models at this point (e.g., different selection criterial, different numbers of predictors, tree parameters, etc.) and perform subsequent runs of CVMC. Once you have selected a model, you can run it on the full data set to obtain a final model fit that can be used to predict future observations.

Learning Curves

An important issue in predictive modeling is having too few samples or too many predictors to generate a meaningful model. Too many predictors can be addressed with predictor reduction techniques. Learning curves are a technique useful in determining if there is an adequate number of samples.

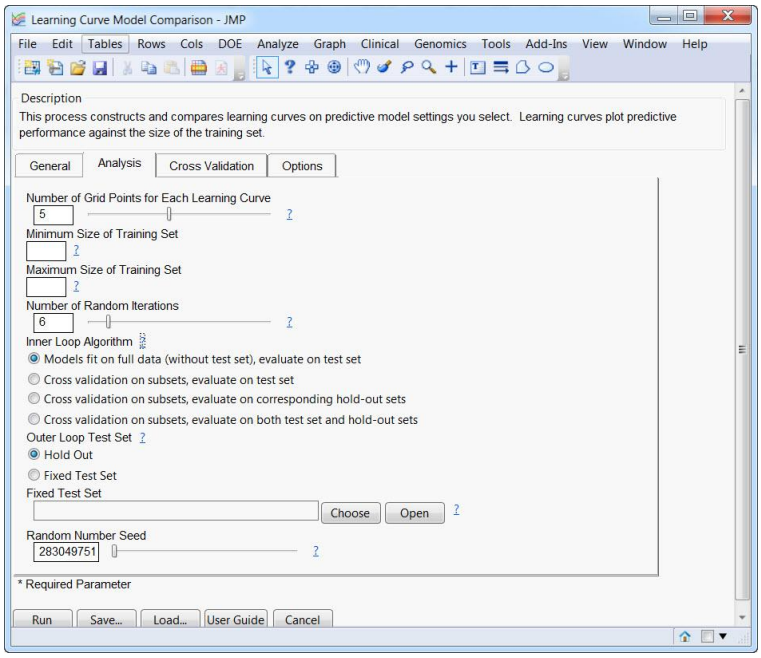
Note: This is a different problem from classic statistical sample size / power calculations. Here we are concerned with how sample size relates to prediction performance. The results will indicate if more samples are likely to help improve performance.

Keep in mind that for every prediction problem there is a theoretical upper bound on performance that is dependent on the phenomenon being predicted. In some applications an AUC of 0.8 might be considered excellent whereas in others 0.8 might be considered very poor. Learning curves can help you determine how close you might be to achieving the best possible predictor.

1. Select **Predictive Modeling > Model Comparison > Learning Curve Model Comparison** from the **Genomics Starter**.

In this example, we will show results from the GLM model with 5 predictors, which was the best model fit from refinement our previous CVMC exercise.

2. The **General** tab is filled out identically to that of the **Cross Validation Model Comparison** window.
3. Click the **Analysis** tab.



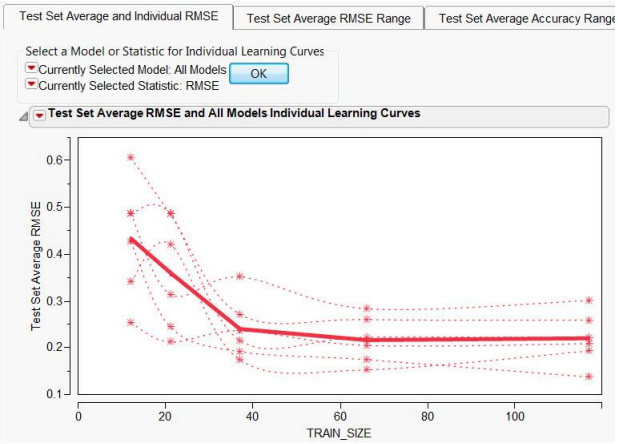
Learning Curve Model Comparison works by performing a series of CVMC runs along a fixed grid of subset sizes of the full data set. In this instance, 5 points will be generated for each learning curve. The points are logarithmically spaced between the maximum and minimum sizes specified, or by default between 1/10 the size of the data set and the size of the full data set.

The **Number of Random Iterations** specifies the number of times to iterate at each point on the grid. So with 5 points on the grid and 6 random iterations, we have 30 total iterations thus far.

The cross validation criteria can be set with the **Inner Loop Algorithm** selections. The simplest is the first choice which mimics standard cross-validation. Further rigor can be added by cross-validating on the subset training sets themselves, then specifying the set for the evaluation set. The test data set the model is fit to can either be from the hold out, or it can be a separate test data set. This latter mode can provide further information concerning the quality of the separate test and training data sets.

4. The **Cross Validation** tab is filled our similarly to that in **Cross-Validation Model Comparison**. Note that the number of iterations specified here multiplies those on the previous tab, so run times can be lengthy.
5. Click **Run** to start the process.

The results show the value for each iteration of the model at different training set sizes. The first pane shows the RMSE.

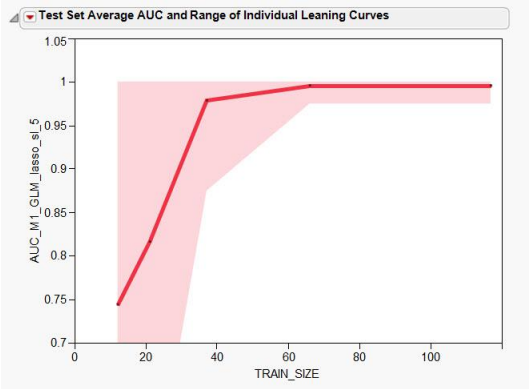


- Click the red triangles above the figure to look at each model individually if more than one model was used, and change the *y*-axis values from RMSE to Accuracy or AUC.

Each dotted line represents one of the iterations and the points along the dotted lines denote the sample sizes. The solid line is the mean of the iterations. What we see is that at small sample sizes, in this case 12 and 21 samples, the RMSE is higher than the final values and there is a large varation between iterations. Ideally for all the calculated values, the variation between iterations will become less and the mean value will stabilize.

A flattening of the learning curve for the upper sizes indicates that additional samples will likely not improve performance. Conversely, if the curve is still sloping on the right hand side, then more samples will likely help.

The subsequent tabs show the range of values of the iterations versus the training size.

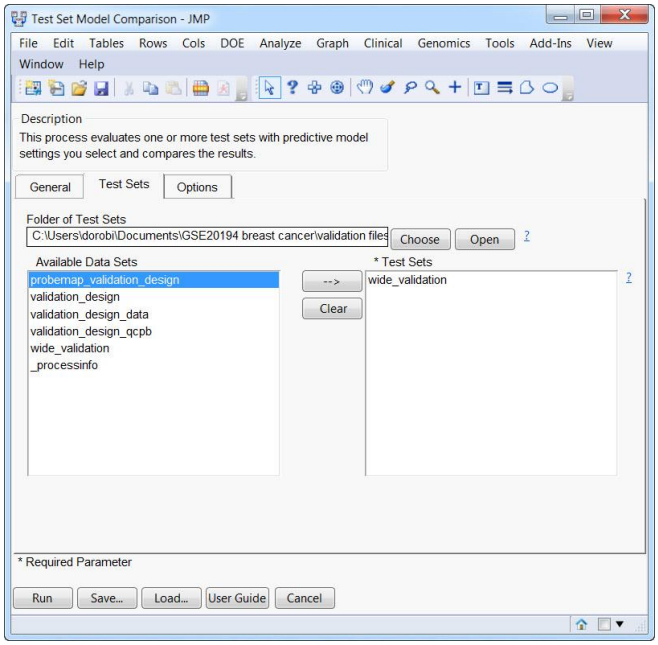


We can indeed see that the range of AUC in this instance is becoming small as the data size increases and the range is small and stable at sizes greater than 60. For this learning curve, it indeed appears that there is a sufficient number of samples in the data set.

Test Set Model Comparison

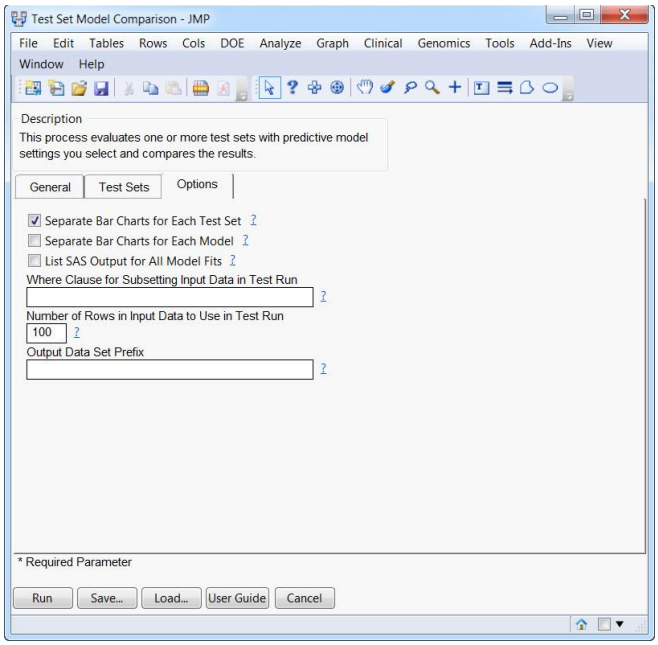
Instead of CVMC, you may have one or more separate test sets against which you wish to compare prediction results of different model fits from a training set. This is a simpler situation than CVMC, as no iterations are required. JMP Genomics enables you to do this in a simple interface.

1. Select **Predictive Modeling > Model Comparison > Test Set Model Comparison** from the **Genomics Starter**.
2. The **General** tab is filled out the same as in **Cross Validation Model Comparison**.
3. Click on the **Test Sets** tab.



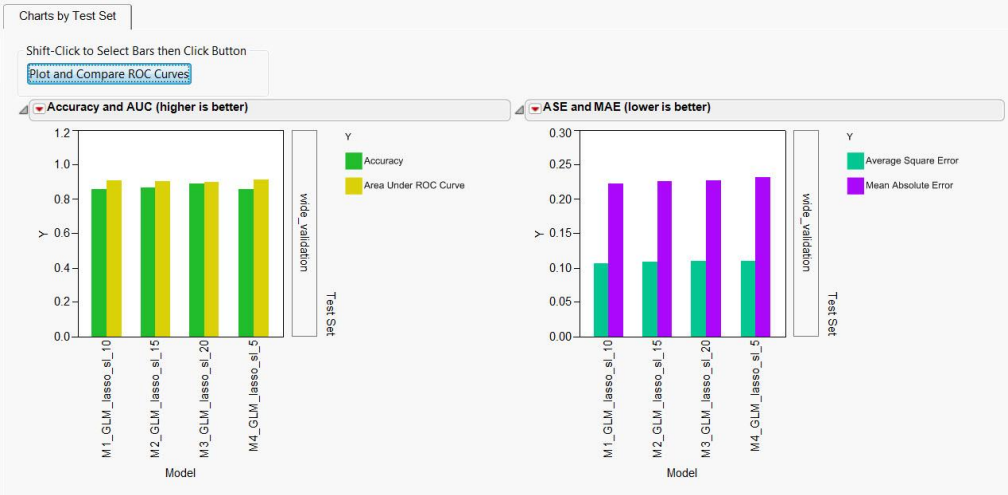
- **Choose** the folder that contains the test set(s). Note that you can select more than one test set for this analysis.

4. Click the **Options** tab.



- In this tab, you can choose how you would like to display the results. **Separate Bar Charts for Each Test Set** is selected by default.

5. Click **Run** to start the process.

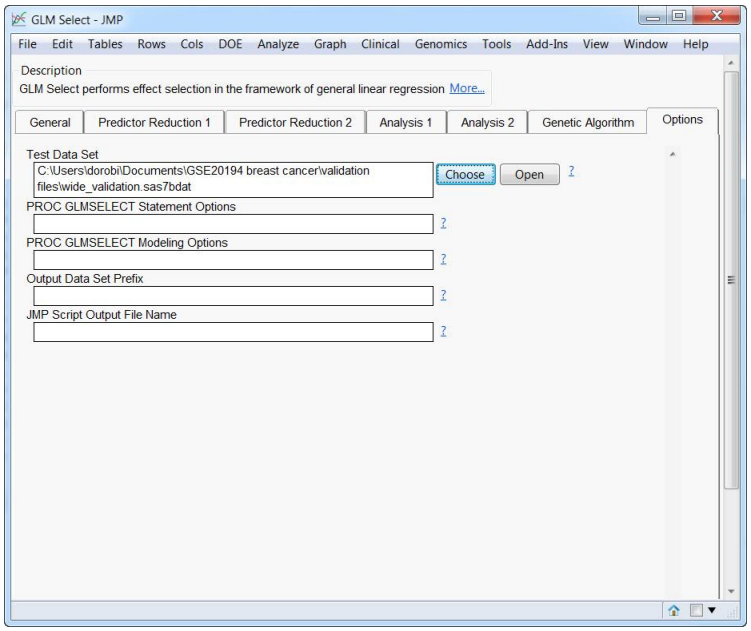


Four analyses are performed for each test set and model. The test sets' Accuracy, Area Under Curve, Average Square Error and Mean Absolute Error are calculated. ROC curves can be generated from the AUC results by highlighting the bars of interest (hold down the Shift key to highlight more than one) and clicking on the **Plot and Compare ROC Curves** button above the chart.

Whole Model Fit, Training versus Test Set Comparisons

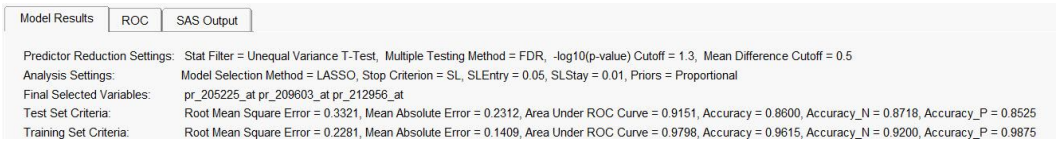
After selection of the final model, a whole model fit can be performed on the training data set, and those parameters applied to the test set. This will assist in determining if the model derived from the training set is generalizable across different analyses.

1. Select **File > Load Genomics Setting** from the JMP menu.
2. Go to the settings file created previously and load in the best model setting.
3. Click on the **Options** tab.

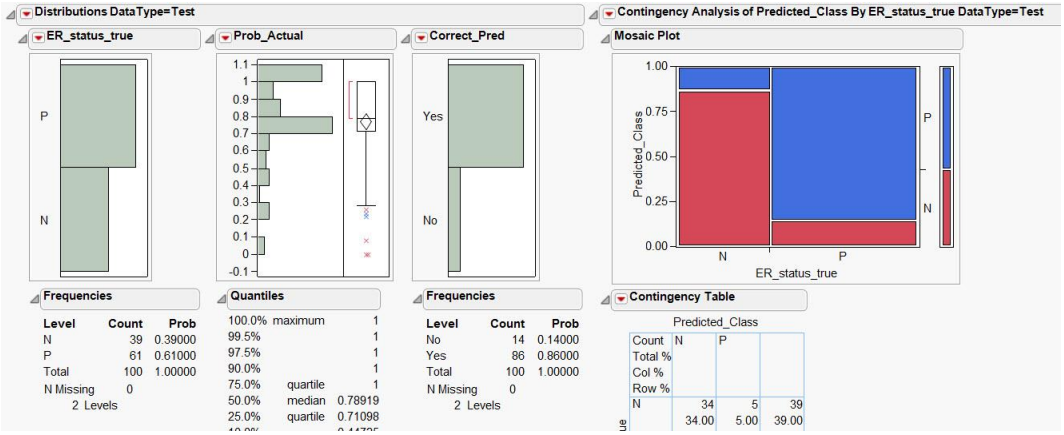


- **Choose** the desired test data set. On any of the Partition Tree processes, a validation set can also be included in this process.

4. Click **Run** to start the process.



The top of the dashboard gives the summary results for the training and the test set. We can see that there are 3 final probes included in the model along with the same model-fit assessments used in the learning curve model comparison. Below the summary, the graphical summaries of each data set are displayed.



From left to right are the distribution of the samples in the data set, the probability of being in the correct group, followed by the distribution of correct vs. incorrect calls. So it is very simple to see that there is a strong trend towards correct predictions and the predictions are correct 86% of the time. The mosaic plot is a graphical representation of misclassification. The values are summarized in the contingency table below the plot, but in this instance we can see that the rate of misclassification is about equal between the ER receptor positive vs. negative groups.

ROC plots and ROC plot statistics are shown on the second tab. A table is generated which contains information on all of the samples so you can easily assess which specific samples are misclassified. For some models (e.g., GLM Select), an additional SAS output frame is generated which includes additional predictor selection results.

Conclusion

JMP Genomics offers a comprehensive set of routines for predictive modeling in a life sciences context. You can simultaneously cross-validate and compare an arbitrary number of model selected from nine different classes of algorithms, each with numerous tuning options. Learning curves help you determine if large sample size will improve prediction performance. You can also compare results against fixed test sets.

References

MAQC Consortium (2010) The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature Biotechnology* 28, 827-838.

Fan, X. et al. (2010) Consistency of predictive signature genes and classifiers generated using different microarray platforms. *The Pharmacogenomics Journal* 10, 247-257.

Oberthuer. A. et al. (2010) Comparison of performance of one-color and two-color gene-expression analyses in predicting clinical endpoints of neuroblastoma patients. *The Pharmacogenomics Journal* 10, 258-266.

Huang, J. et al. (2010) Genomic indicators in the blood predict drug-induced liver injury *The Pharmacogenomics Journal* 10, 267-277.

Miclaus, K. et al. (2010) Variability in GWAS analysis: the impact of genotype calling algorithm inconsistencies. *The Pharmacogenomics Journal* 10, 324-355.

Miclaus, K. et al (2010) Batch effects in the BRLMM genotype calling algorithm influence GWAS results for the Affymetrix 500K array. *The Pharmacogenomics Journal* 10, 336-346.

Zhang, L. et al (2010) Assessment of variability in GWAS with CRLMM genotyping algorithm on WTCCC coronary artery disease. *The Pharmacogenomics Journal* 10, 347-354.

Hong, H. et al (2010) Assessing sources of inconsistencies in genotypes and their effects on genome-wide association studies with HapMap samples. *The Pharmacogenomics Journal* 10, 364-374.

Acknowledgements

Several members of the JMP Life Sciences Team continue to participate in MAQC data analysis efforts and are co-authors on several prominent MAQC Phase II publications. Others on the team have provided critical support over the years in development of JMP Genomics predictive modeling capabilities. Members include (in alphabetical order): Wenjun Bao, Tzu-Ming Chu, Shannon Conners, Wendy Czika, Jordan Hiller, Lili Li, Geoff Mann, Stan Martin, Kelci Miclaus, Valerie Nedbal, Padraic Neville, Tom Pedersen, Doug Robinson, Susan Shao, Pei-Yi Tan, and Russ Wolfinger.