Virtual Beach v 2.4 User's Guide

Mike Cyterski, Mike Galvin, Kurt Wolfe, and Rajbir Parmar

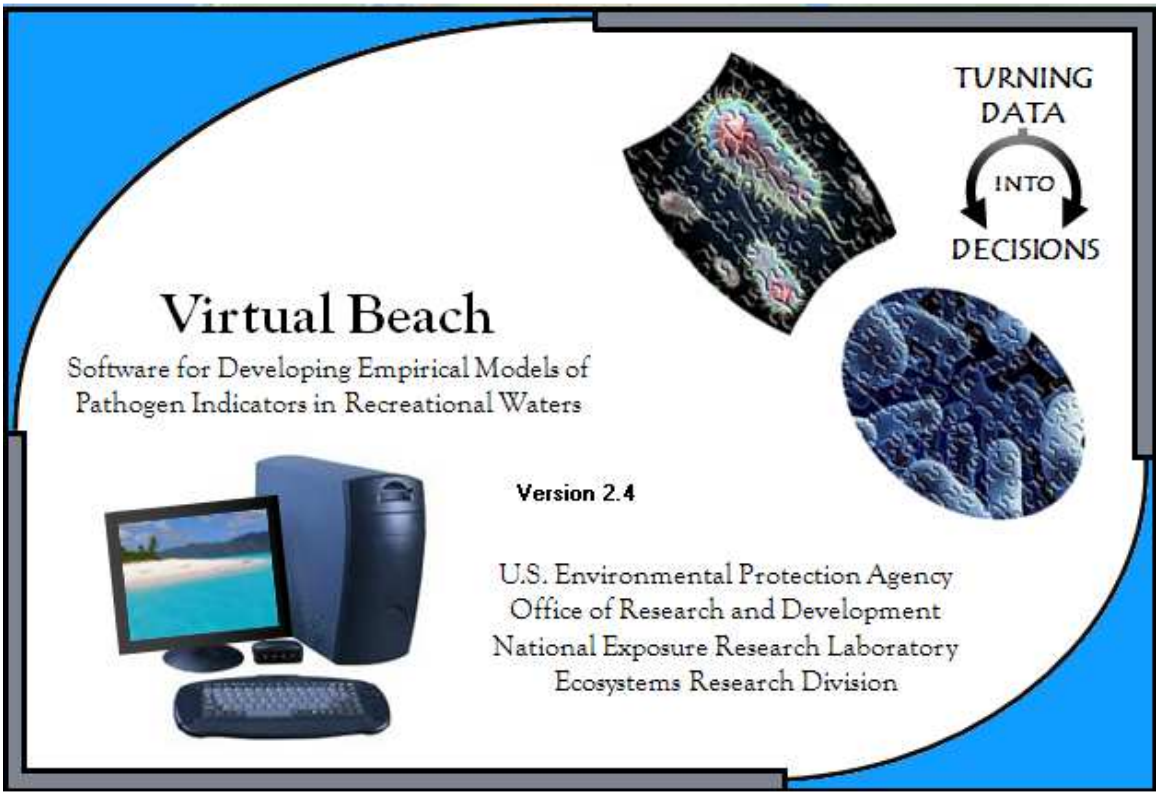


Table of Contents

List of Figures

1. INTRODUCTION

Virtual Beach version 2.4 ($VB_{2.4}$) is a decision support tool. It is designed to construct site-specific Multi-Linear Regression (MLR) models to predict pathogen indicator levels (or fecal indicator bacteria, FIB) at recreational beaches. MLR analysis has outperformed persistence models (using the most recent FIB concentration as the sole predictor of the next FIB concentrations, i.e., $y_t = y_{t-1}$) at beaches where conditions, such as weather, water conditions, and human and animal traffic levels, change significantly from day to day (Frick, Ge et al. 2008).

1.1 On Predictive Modeling

In any predictive modeling endeavor, variability and uncertainty are always associated with model output, arising from a variety of reasons that are impossible to eradicate completely from the modeling exercise. $VB_{2.4}$ attempts to be forthright with this fact by issuing a probability of exceedance for any regulatory standard that the user wishes to investigate. Even so, there is no guarantee than every model prediction will be correct, and a situation where the model predicts water quality to be good enough for public recreation might be erroneous. Decisions to allow or not allow swimming at beaches must be made, however, and in the best case scenarios the regression models developed with $VB_{2.4}$ will outperform less rigorous predictive efforts.

1.2 Recommended User Background

$VB_{2.4}$ is our attempt to create a decision support software tool that will assist someone with little statistical knowledge in developing a multiple linear regression model based on their available data. Some familiarity with regression modeling and residual analysis will no doubt benefit a $VB_{2.4}$ user, although we believe that, after only a few sessions, someone with very little background in statistics can produce defensible regression models using $VB_{2.4}$. We note that these MLR models, or any other statistical models, will only be as effective as the data used to develop them. No statistician, however skilled, can turn a dataset filled with worthless independent variables (i.e., IVs) into a useful predictive device.

$VB_{2.4}$ has four major components:

- Beach location map interface where users can locate their site, define the orientation of the beach, and examine nearby potential data sources.
- Data processing spreadsheet interface that facilitates the import and manipulation of MLR model variable data.
- Modeling interface presenting options for performing MLR analyses, including a residuals component to examine regression residuals, allow optional elimination of highly influential data records, and perform recalculation of the regression model.

- Prediction interface allowing entry of new data and subsequent estimation of pathogen indicator levels using a selected MLR model.

Each component is accessible from the application's main window via selectable tabs. The Beach Location and Data Processing tabs are always visible, the Modeling tab becomes visible once the input data have been validated, and the MLR Prediction tab appears when model-building is complete and a model is selected.

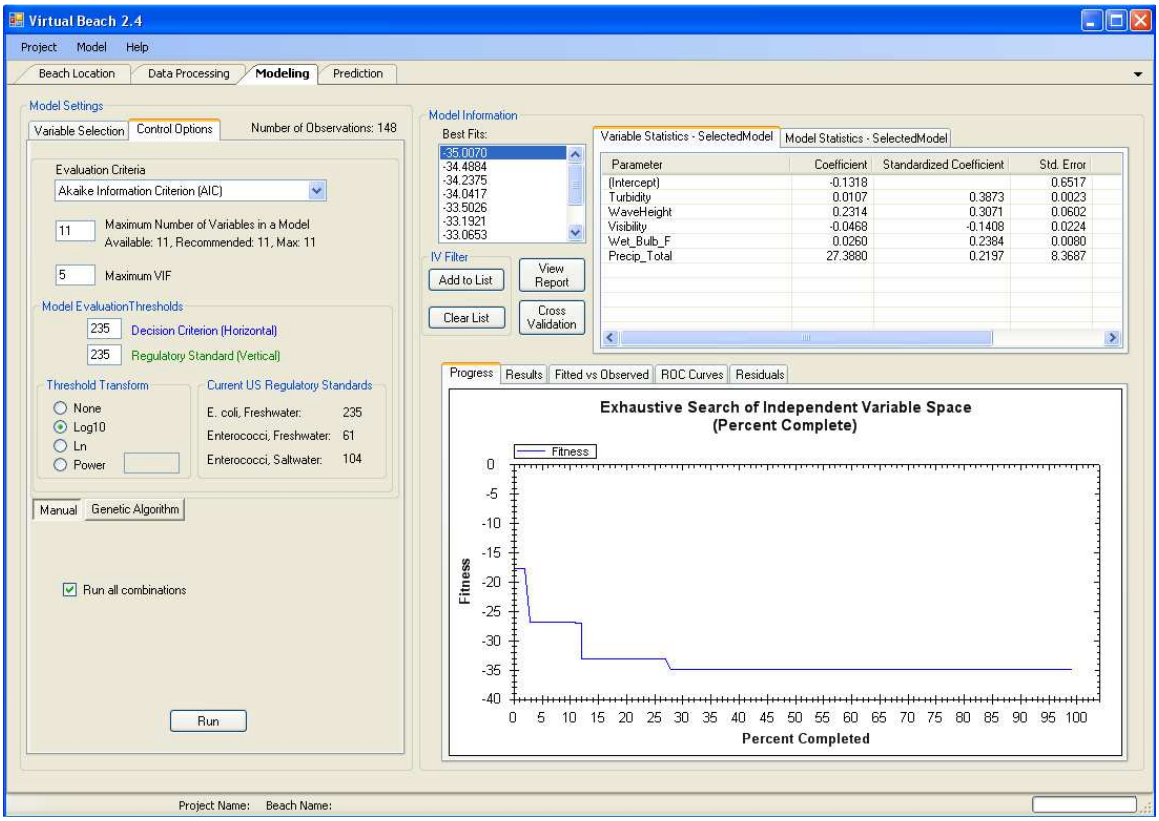


Figure 1. The four major component tabs of $VB_{2.4}$ - the modeling tab is currently active

1.3 History and Comparison of $VB_{2.4}$ to Earlier Versions

$VB_{2.4}$ is derived from the Virtual Beach Model Builder application ($VB_{1.0}$) developed by Walter Frick and Zhongfu Ge. $VB_{1.0}$ can be characterized as a MLR model-building tool that supports a primarily manual analysis of data sets via visual inspection of data plots and manipulation of variables (e.g., transformations, creating interaction terms), followed by an iterative process of testing, comparing and evaluating models. The fitness of developed models is computed and tracked, allowing for comparison and eventual selection of a "best" model for the dataset under consideration. This model can then produce estimates of pathogen indicator levels using current or forecasted environmental data from the site.

$VB_{2.4}$ enhances the functionality of its predecessor, performing similar functions (visual inspection of univariate data plots, manual transformations of individual variables, MLR model building, prediction, etc.), but also automating and extending functionality in several ways:

- The Map component provides users with information on the location and availability of local data sources (NWIS/NCDC data) through the map interface. These sources can provide recently collected and/or forecasted data for generating predictions by a chosen MLR model.

- The Map component provides a convenient method for defining beach orientation by overlaying the beach on current shore-line layers (satellite images, Google Maps, MS Virtual Earth, etc). Given this orientation, $VB_{2.4}$ can calculate wind, wave, or current components (A component is parallel to shore and O component is perpendicular to shore), which can be important predictor variables.

- Although manual processing and analysis of imported data (visual inspection of univariate data plots and the transformations/interactions of variables) has been retained, the Data Processing component of $VB_{2.4}$ provides automated generation of all possible 2nd order interaction terms amongst a set of IVs, formation of more complex functions of multiple columns, and automated testing of a suite of variable transformations for improved model linearity. This functionality increases the number of models to evaluate during later selection routines and removes the burden/difficulty of manual assessment placed on users of $VB_{1.0}$.

- Multi-collinearity amongst predictor variables is handled automatically in the Model Building component. Any model containing an IV with a high degree of correlation with other IVs (as measured by a large Variance Inflation Factor -VIF) is removed from consideration during model selection. The VIF threshold is user-defined with a default value of 5.

- During model selection, MLR models are ranked by a user-selected evaluation criterion. Possible criteria include R^2, Adjusted R^2, Akaike Information Criterion (AIC), Corrected AIC, Predicted Error Sum of Squares (PRESS), Bayes Information Criterion (BIC), Accuracy, Sensitivity, Specificity, or the model's Root Mean Square Error (RMSE). Regardless of which criterion is chosen, the software records the ten best models in terms of that criterion. In comparison, $VB_{1.0}$ had only a single comparative criterion, Mallow's Cp.

- As the number of IVs in a dataset increases, possible MLR models increase exponentially (considering transforms/interactions), resulting in trillions of possible models from a modest number (12-13) of IVs. $VB_{2.4}$ implements a Genetic Algorithm (GA) that effectively and efficiently searches for the best possible MLR model. Alternatively, $VB_{2.4}$ users can perform an exhaustive calculation in which all possible combinations of IVs are used and tested if the number of possible models is reasonably small (circa 100,000). Both the GA and exhaustive approaches greatly expand the model-building capabilities of $VB_{2.4}$, compared to $VB_{1.0}$.

- Users no longer have to enter data values in transformed, interacted, or component-decomposed form to make a prediction with a chosen MLR model. On the $VB_{2.4}$ Prediction tab, a user-selected model is coded into an input grid with

data entry columns matching the model's main effects. Any mathematical manipulation of these IVs is then automatically performed prior to making predictions.

2. INSTALLATION AND EXECUTION

$VB_{2.4}$ is developed with MS Visual Studio 2010, written in C#, using multiple public domain system components (Weifen Luo Docking UI, ZedGraph, and GMap.Net) and employs a single licensed statistical library (Extreme Optimization). No license or software purchase is required by the user to install and run the application, but an internet connection is required to display maps. Users must have Microsoft XP or Windows 7 OS with the DotNet Framework 4.0 to assure proper installation and operation. Assorted errors have occurred when running Windows Vista OS. Certain $VB_{2.4}$ data manipulation and model-building operations are computationally intensive so faster CPUs are better, but most new laptops or desktop systems will be adequate. Disk space requirements are modest (less than 20 MB) if the DotNet Framework is installed; if not, the Framework installer requires ~ 175 MB of disk space. The $VB_{2.4}$ application installer will attempt to download and install the DotNet Framework 4.0 if it is not installed on the target system; this also requires a network connection. If necessary, a user can freely obtain the DotNet Framework 4 installer at:

http://www.microsoft.com/download/en/details.aspx?id=17851

The EPA's Center for Exposure Assessment Modeling (CEAM) web site distributes VB at:

http://www2.epa.gov/exposure-assessment-models/virtual-beach-vb

Obtain and initiate execution of the $VB_{2.4}$ application installer and follow the on-screen instructions. After installation, a shortcut will appear on your desktop to start the software.

2.1 Viewing this Documentation

The $VB_{2.4}$ user's guide can be accessed within the software via the top-level Help/ User Guide menu selection or in a context-sensitive fashion via the F1 key. Invoking F1 will launch Adobe Acrobat or Adobe Reader (if installed) and open the user's guide to the appropriate page. Note that if the guide is already open, the F1 key will have no effect; users must close Reader (or Acrobat) for F1 to launch and open to the correct page. Or if the guide is already open, users can navigate to the area of interest via the Table of Contents. The user's guide (Virtual_Beach_24_User_Guide.pdf) can also be opened independently of program operation; it resides within the Documentation folder of the program's installation folder.

3. OPERATIONAL OVERVIEW

Our goal was to make $VB_{2.4}$ straightforward to operate: it is categorized into four functions, each with its own component or interface:

Beach Location – a mapping tab whose utility is meant to provide a basis for generating orthogonal (alongshore and offshore/onshore) wind, current, and/or wave components for the beach under consideration; its use is optional. Such components can be powerful predictors of pathogen indicator levels at the beach, so using the beach definition component is recommended if the dataset under consideration contains wind, wave or current data. This tab is also useful for locating nearby NWIS/NCDC climate and water quality data sources for a specific location.

Data Processing – a spreadsheet tab to support data manipulation procedures on an imported dataset. In addition to wind/current/wave component generation, users can generate new independent variables that represent the products, means, sums, minimums, and maximums of other IVs, as well as common data transformations for the IVs. Statistical indicators help users select the best IV transformations in MLR model-building.

Modeling – this tab allows selection of any eligible IVs for consideration in MLR model-building and model-generation. Model-generation is accommodated by user-selected model evaluation criteria and automatic generation of the ten best-fit models from a search in which all possible combinations of predictor variables are tested, or via a heuristic searching algorithm (the Genetic Algorithm or GA). Regression fit and model variable statistics are generated to help evaluate the usefulness of predictive variables and overall fit. Time series and XY scatter plots, as well as reports on best-fit models, can be viewed and/or saved for further analysis and recording. There are also subtabs for residual analysis on the modeling tab. Here the user can examine plots of a model's regression residuals, including their normality statistics, and eliminate highly influential data records and recalculate the regression model. Altered data sets can be exported for external use and rebuilt models can be selected for the prediction tab.

Prediction -- this tab is comprised of three grids where users can enter or import the needed IVs for the chosen model, enter or import observations that will be compared to model predictions, and examine model predictions and exceedance probabilities. Time series and XY scatter plots of observations versus predictions are shown to help users gauge model effectiveness.

4. PROJECT MANAGEMENT

Oftentimes the user will put an imported dataset through lengthy pre-processing to prepare it for analysis. To avoid repeating all of this work, "project" files can be saved and re-opened via the Project → Save and Project → Open menu selection. Subsequent opening of a saved project file will load the processed data sheet and information on the Beach Location tab, including the beach orientation if the user had defined it. However, no modeling information is saved inside a project file.

In addition to project files, "model" files can be opened and saved using choices under the "Model" menu at the top of the $VB_{2.4}$ interface. A model file contains information on the IVs, regression parameters, and other metadata for the currently selected model in the Modeling or MLR Prediction tab. Whenever a model file is saved, $VB_{2.4}$ will prompt the user to enter a Decision Criterion (DC), Regulatory Standard (RS) and Threshold Transformation for the model. These parameters will be used as initial values (they can be changed when the model file is opened) for later calculations of model sensitivity and specificity, which depend on the numbers of false negative and false positive model predictions (see Sections 7.6 and 7.7).

When users open a previously saved model file from within $VB_{2.4}$, they are taken directly to the MLR Prediction tab where they can use the saved model to generate predictions. Model files are designed for situations where a statistically-savvy developer is charged with developing regression models for a number of beach sites. After the developer chooses a "best" model for a site, the model file can be saved and then delivered to the beach manager who will not use $VB_{2.4}$ for full-scale model development, but only to input new data, generate predictions, and make decisions regarding swimming advisories.

5. BEACH LOCATION MAPPING INTERFACE

On $VB_{2.4}$ application startup, the map interface is shown, but users can go directly to the Data Processing tab if desired.

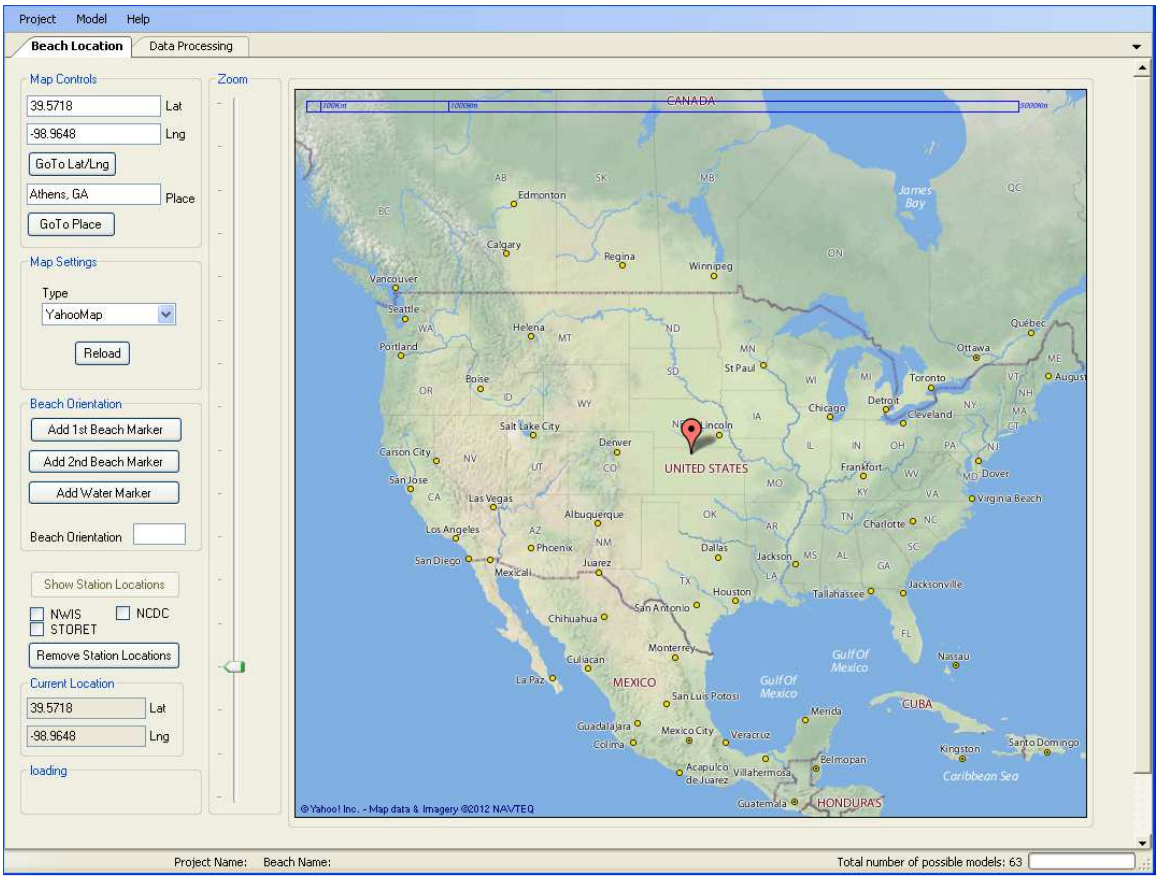
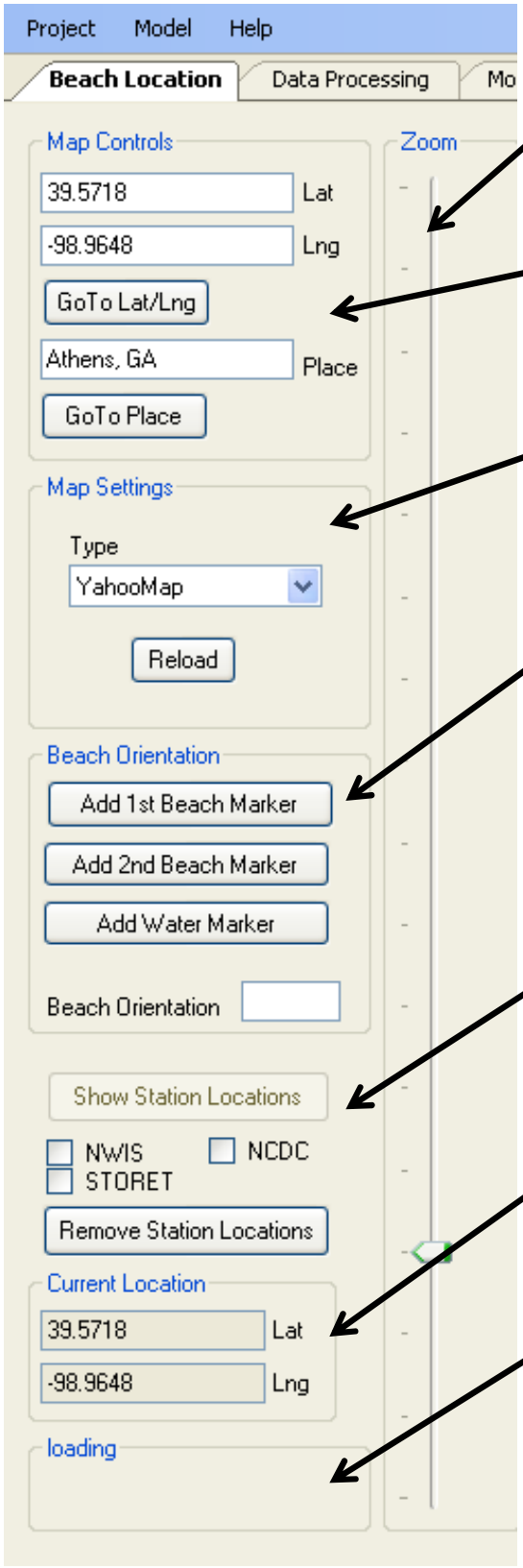


Figure 2. Beach Location interface - the default map type is Yahoo Map, but users have many mapping options

5.1 Finding a Location

The map interface provides map controls that allow users to look up a location manually by panning and zooming (mouse drag on the map and use of the mouse wheel or zoom control). Alternately, a decimal latitude/longitude or place name can be entered. The control uses Google Maps' reverse geo-coding network service to find locations.

Map Controls



Zoom Slider – drag slider up and down to zoom in and out, respectively.

Map Controls – Add Lat/Long and click "GoToLat/Long" button or enter a Place and click "GoToPlace."

Map Settings – Select map type from drop down menu to change the display in the map window.

Beach Orientation – use buttons to add or remove markers on the map. Once the beach shoreline is delineated by placing the 1st and 2nd beach markers, click in the water and then click "Add Water Marker," which will lead to the correct orientation angle being placed into the "Beach Orientation" box.

Show Station Location – if zoomed in enough, select a station type and then click "Show Station Locations" to display such stations on the map.

Current Location – click anywhere on the map to display that points Lat and Long.

Loading – map loading progress bar that shows network download activity for map images.

Figure 3. Beach Location tab controls and their function

5.2 Defining the Beach Orientation

Map control allows delineation of a beach on the map to ascertain its orientation, which is useful if wind, wave, and/or current flow components are to be used in MLR model-building. Maps, as opposed to satellite or hybrid images, provide less shoreline detail so it is recommended that the map setting type use a hybrid or satellite image prior to adding point locations that define beach boundaries. Once displayed, click on the map (a red marker will appear) and select the "Add 1st Beach Marker" button; this represents the first point of the extent of your beach shoreline. Repeat this for the second beach marker and click on the map to indicate which side of the shoreline represents the water; then hit the "Add Water Marker" button. Marker points will turn green as you add them. Once the water marker is added, a shaded box (the beach) appears and the computed orientation angle will be displayed.

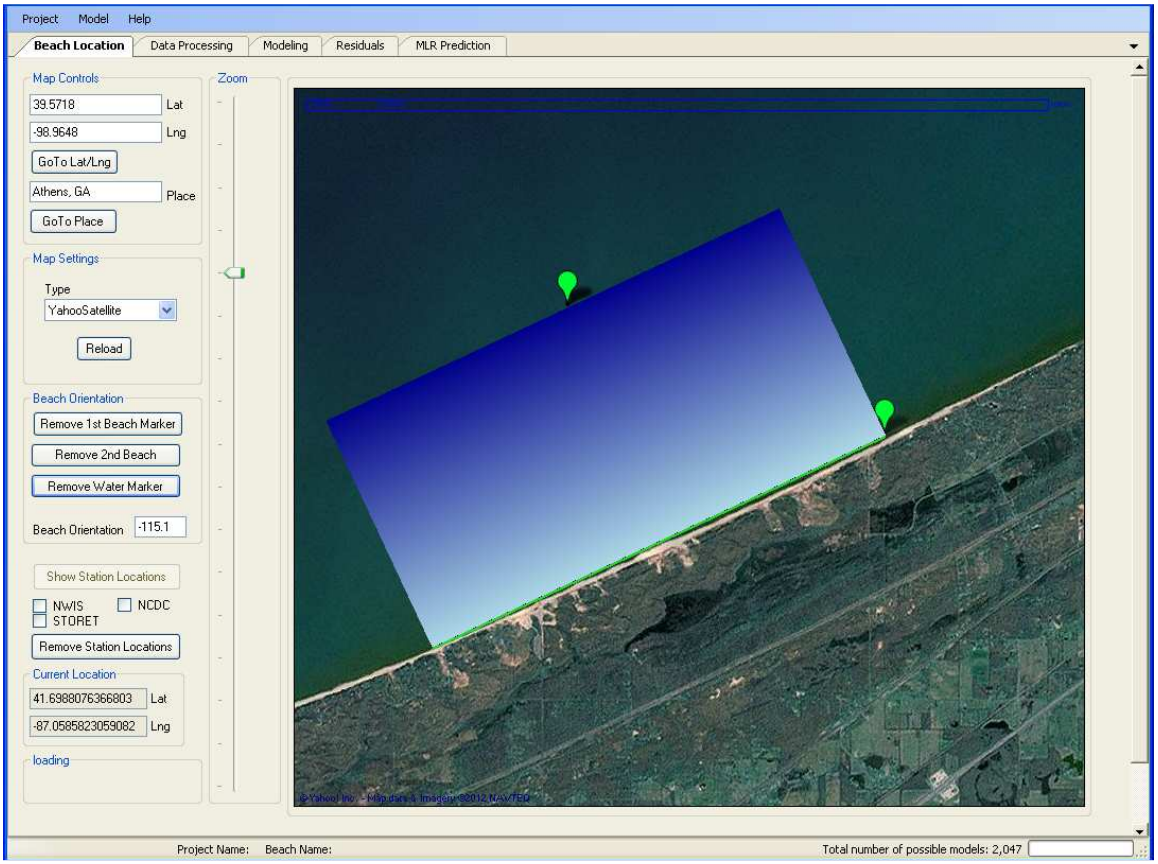


Figure 4. Adding shoreline and water markers to define beach orientation

Points can be added or removed until the user is satisfied with the beach representation. To recall the computed beach orientation in the data processing components creation screen (see Data Processing section below), users can either save and then re-open a project file or they can note the beach orientation on the mapping screen and manually enter that angle on the components calculation screen.

5.3 Finding nearby Water Quality, Flow, and Climate Information Sources

Possible nearby data sources for the area of interest may be located and displayed on the map. USGS NWIS and NOAA NCDC station markers at a zoomed-in map area can be located and displayed by checking appropriate items in the map window and clicking the "Show Station Locations" button. Note that the "Show Station Locations" button is only enabled when zoomed-in to an appropriate level (e.g., zoom level three as measured from the top of the zoom control slider). If either of the selected station categories (NWIS and/or NCSC; the STORET station category, although present on the control, is not yet functional) are present within the map display area, they will appear. Also note that the network server that produces NCDC station locations restricts location requests to one every 30 seconds – a one-half minute delay is required for subsequent location requests and an error message will be displayed if the appropriate wait time has not elapsed. Once station location markers are displayed on the map, hovering over the top-left hand corner of any station marker will display station ID information. With that information, users can visit the appropriate web address to gather water/weather data for the area of interest.

Figure 5. NOAA/NCDC station marker showing station ID information

Figure 6. USGS/NWIS station marker showing station ID information

USGS NWIS web site URL: http://waterdata.usgs.gov/nwis/inventory
NOAA NCDC web site URL: http://www.ncdc.noaa.gov/oa/climate/stationlocator.html

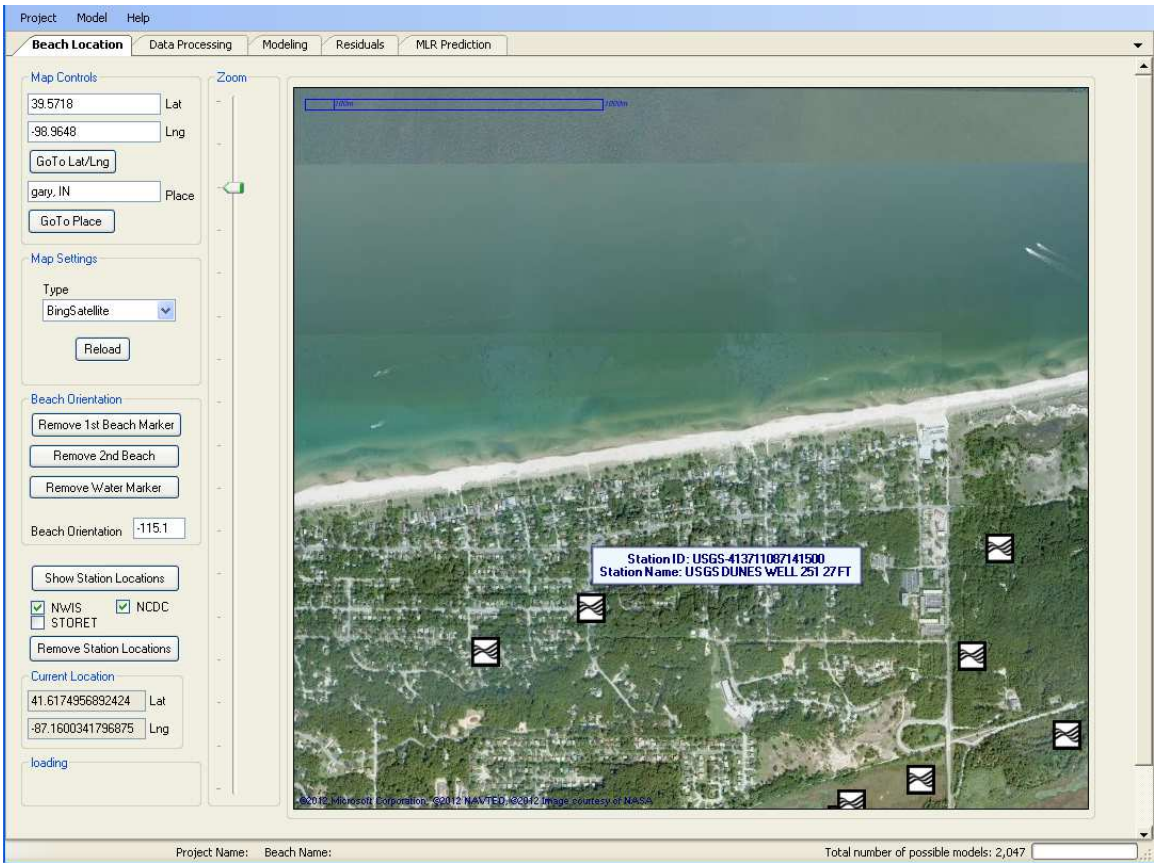


Figure 7. Beach Location interface showing station markers near Gary, Indiana

5.4 Saving Beach Information in a Project File

Use the Project→Save menu bar selection to open a Save File dialog and to save the project information to disk. Beach marker and angle information is saved in the file name provided; the saved file can be anywhere, but using the "Project Files" folder (found in the $VB_{2.4}$ root install folder) is recommended.

6. DATA PROCESSING

6.1 Data Requirements and Considerations

$VB_{2.4}$ accepts files from Excel 2007 or earlier (Excel 2010 is not currently supported), as well as comma-separated-value (CSV) text files. Input data must conform to certain standards:

- The first row of any data column must be a header with the IVs name. For best operation of the software, the column name should be composed of letters, numbers (however, don't begin the column name with a number), and/or underscores, i.e., "_". Other characters in column names can cause problems.
- The first (left-most) column of the dataset must be identification for the observations, typically a date or time stamp that indicates when the observation was collected. The only requirement is that each row MUST have a unique ID. $VB_{2.4}$ will not import datasets with non-unique IDs in the first column. If the first column is a time stamp, $VB_{2.4}$ plotting functions will work best if the column is in chronological order, from earliest to most recent observations.
- The second column of the dataset will initially be set as the dependent or response variable; however, this can be changed after data are imported. Any subsequent columns will be considered to be IVs.
- Variable measurement units are not considered, but certainly affect predictions. Make sure any data used for predictions are in the same units as those used to build the models; for example, do not build a MLR model with water temperature in degrees Fahrenheit, then later import water temperature in degrees Celsius for predictions. It is prudent to include unit information in the column names (e.g., WaterTemp_C) to remind the user of the proper units when making predictions.
- Missing data (blank cells) are permitted on import, but must be dealt with in Data Processing prior to modeling.
- If present in the imported Excel data sheet (other than in column names or the first ID column), cells with non-numeric values (i.e., symbols or text) are turned into empty cells. If such non-numeric characters are present in an imported .csv file, they will be imported to the data grid, but will be recognized as anomalous data during the required validation scan and will have to be dealt with (deleted or turned into a numeric value) at that time.
- $VB_{2.4}$ recognizes any column of data with only two different values as categorical. If you have a column of categorical data with more than two values, you can designate it as categorical, using methods described below. The ramification of a variable being identified as categorical is that $VB_{2.4}$ leaves it out of transformation processes.
- $VB_{2.4}$ will automatically disable, but not delete, any column containing only a single value upon import. These columns are essentially useless for predictive purposes.
- There is no hard-coded limit on the number of IV columns one can import; however, a practical limit exists that depends on system processing resources.

There is also an inherent limit: - documentation indicates that the grid components used in the application are designed for a maximum of 300 columns before performance issues degrade the application. Modeling 250+ columns of data presents circa $2(10)^{20}$ possible data combinations for MLR processing. The Genetic Algorithm handles this modeling task, but choosing "Run all combinations" would likely take an immense amount of time to complete. Depending on how many additional IVs will be created by the user, importing a dataset with less than 100 IVs should be acceptable.

6.2 Importing a Dataset

When users first click on the Data Processing tab, they open a dataset using the "Import" button. This brings up a dialog screen where a directory explorer can be used to find the data file and open it. If the dataset is an Excel file with multiple sheets, a dialog box opens to ask the user which to import.

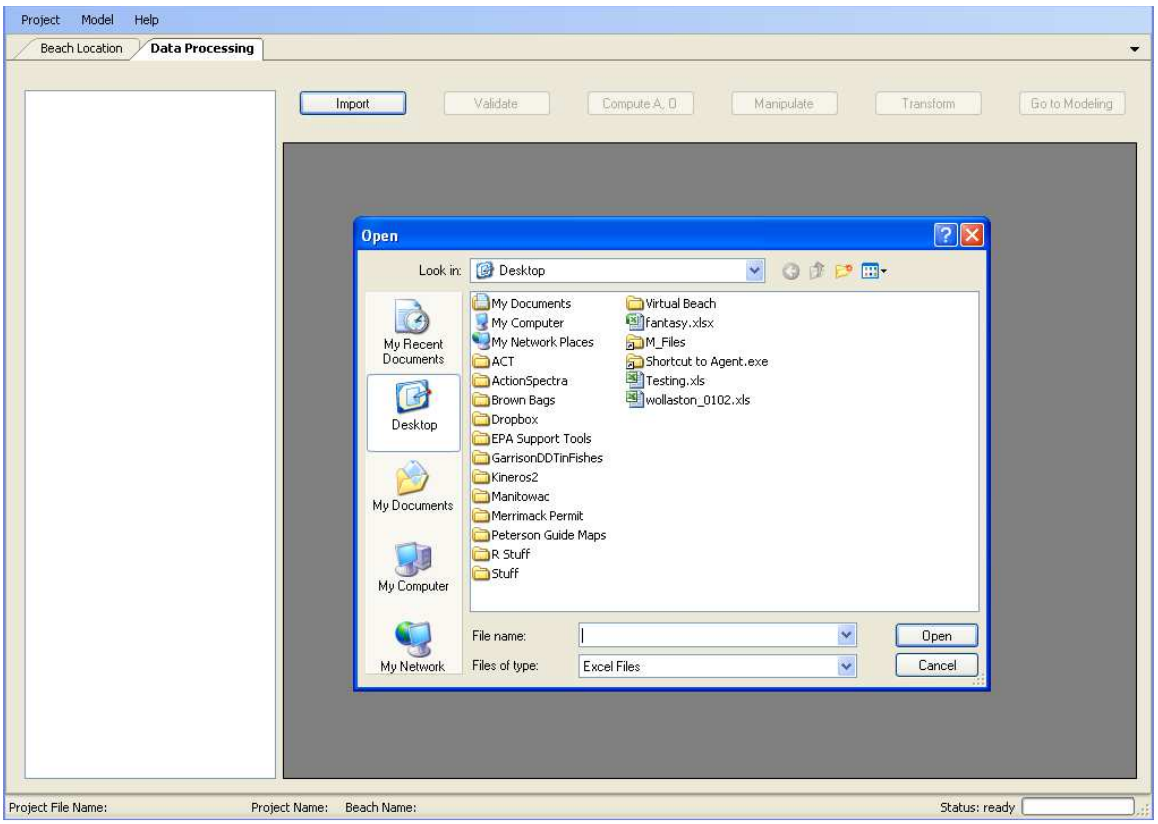


Figure 8. Importing a dataset into the Data Processing tab

Once imported, the data grid is shown as a spreadsheet on the right. The second column of the spreadsheet will be highlighted in blue to indicate its status as the current response variable. Information about the dataset, such as number of rows and columns, name of the ID column and name of the response variable, appear on the left. At this point the grid cannot be edited or interacted with in any manner; to access additional processing functionality, the data must be validated.

6.3 Validating the Imported Data

The "Validate" options window can be accessed by clicking the "Validate" button at the top of the Data Processing tab. This window primarily launches a required data scan to identify blank and non-numeric data cells in the imported spreadsheet. However, one can also find and replace other specified values (e.g., a missing data tag like -999) in the dataset using the "(Optional) Find:" input box.

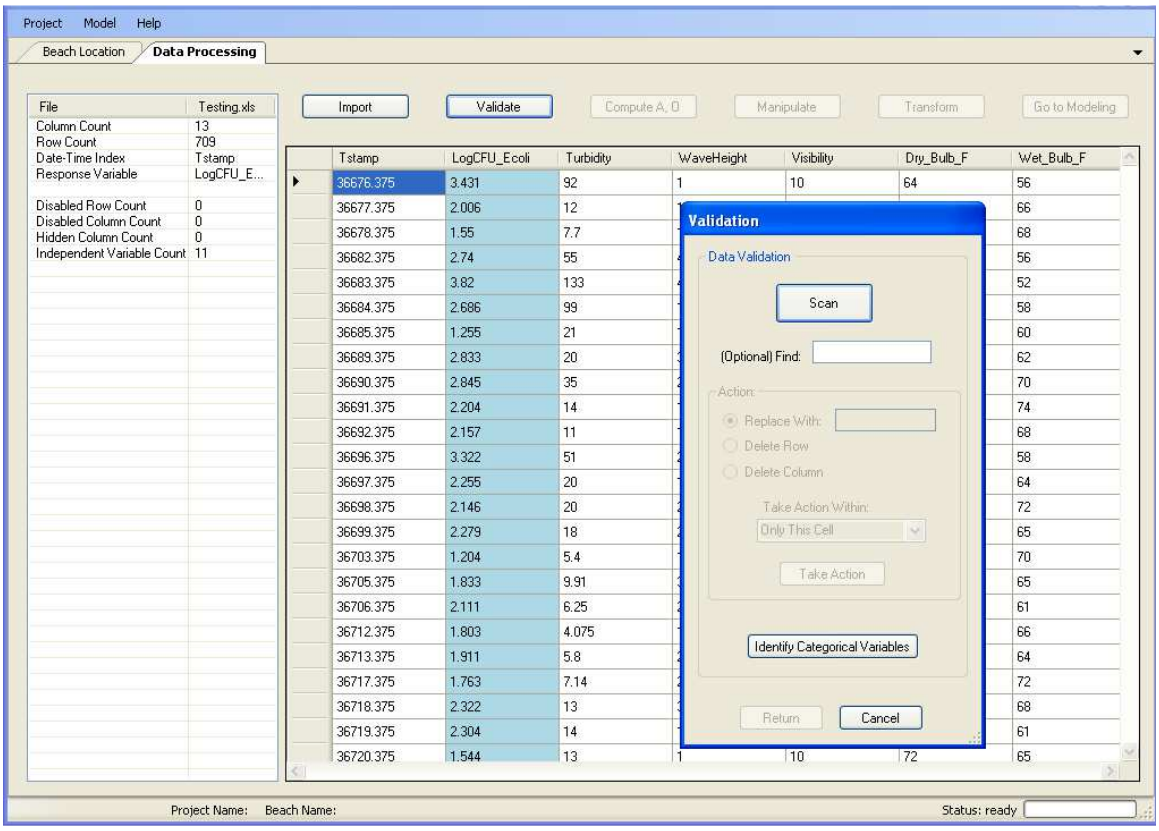


Figure 9. Data validation required to begin data processing

To validate the data, the user clicks "Scan." $VB_{2.4}$ then goes through the spreadsheet, cell by cell, looking for blanks, non-numeric, or user-specified values entered in the "Find:" input box. If one of these types of cells is found, the scan will stop to highlight that cell. Users must decide how to deal with the cell using choices in the "Action" section: they can replace the bad cell with a specified value, using the "Replace With:" input box, or they can delete the row or column containing the bad cell. The user must decide where to implement the chosen action with the "Take Action Within" menu. Possible choices are "Only this Cell," "Only this Row," "Only this Column," "Entire Row," "Entire Column," and "Entire Sheet." Items in this menu are context-sensitive, i.e., they change depending on which Action is selected. This setup gives the user flexibility, for example, to delete all rows containing missing values within one specific column of data (Action would be "Delete Row" taken within the "Entire Column"), and replace all missing values with a user-specified numeric value within another column of data (Action would be "Replace With:" taken within "Entire Column"). The cell, row, and column reference will always refer to the highlighted cell. After setting the "Take

Action Within" menu, the user clicks the "Take Action" button, $VB_{2.4}$ makes the specified changes to the spreadsheet, and the scan continues. When the entire spreadsheet has been scanned and all bad cells have been fixed, $VB_{2.4}$ reports that "no anomalous data have been found," and the user can click the "Return" button to close the Scan window.

As stated earlier, $VB_{2.4}$ will not attempt to transform categorical data columns. It automatically identifies columns with only two unique values as categorical, but if the user has other categorical IVs with more than two categories, those should be identified to $VB_{2.4}$ by the "Identify Categorical Variables" button.

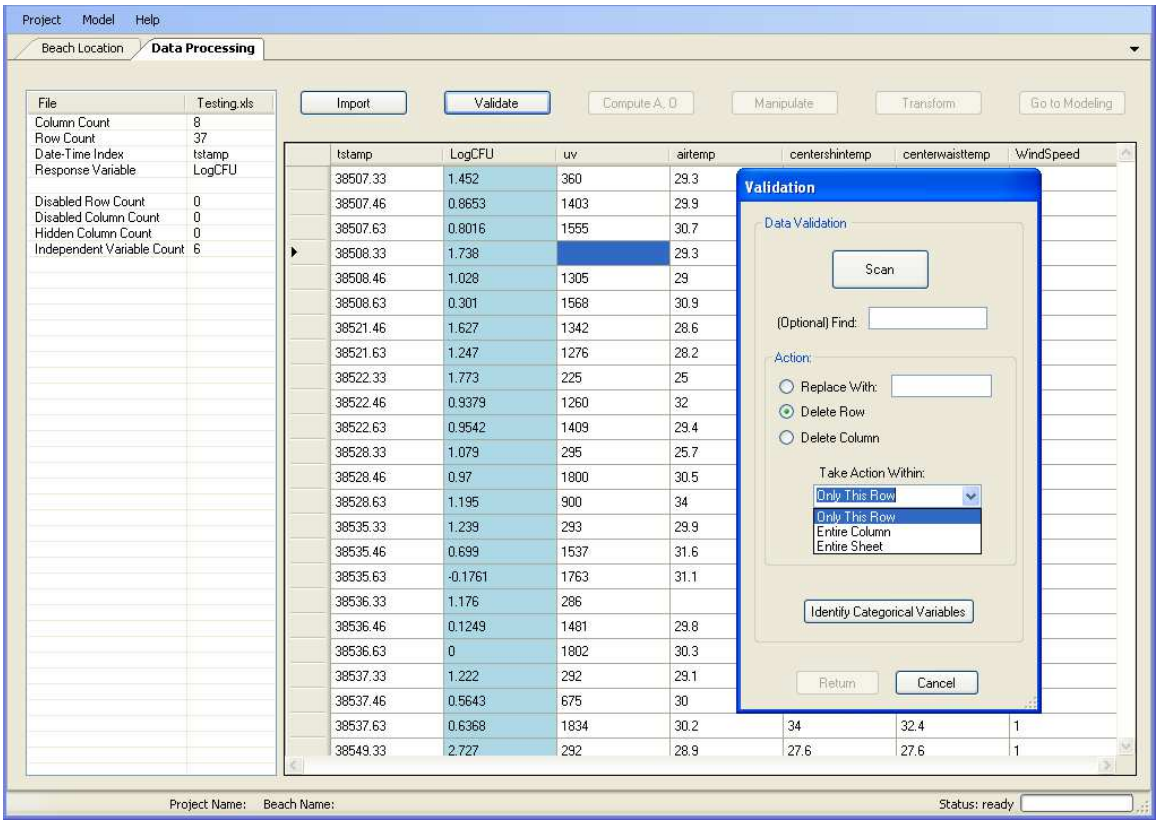
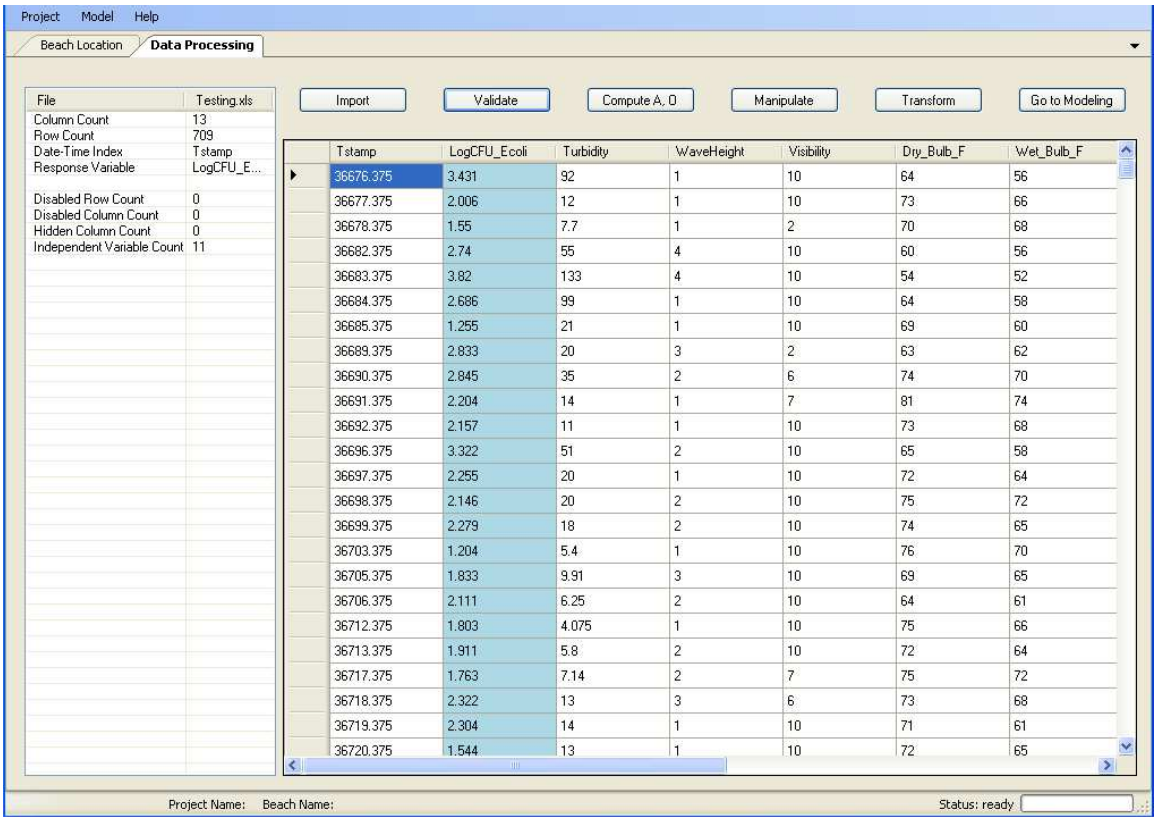


Figure 10. Context-sensitive choices for the "Take Action Within" drop-down menu

6.4 Working with a Dataset Post-Validation

After the dataset has passed the validation scan, the function buttons across the top of the Data Processing tab are enabled.



Tstamp	LogCFU_Ecoli	Turbidity	WaveHeight	Visibility	Dry_Bulb_F	Wet_Bulb_F
36676.375	3.431	92	1	10	64	56
36677.375	2.006	12	1	10	73	66
36678.375	1.55	7.7	1	2	70	68
36682.375	2.74	55	4	10	60	56
36683.375	3.82	133	4	10	54	52
36684.375	2.686	99	1	10	64	58
36685.375	1.255	21	1	10	69	60
36689.375	2.833	20	3	2	63	62
36690.375	2.845	35	2	6	74	70
36691.375	2.204	14	1	7	81	74
36692.375	2.157	11	1	10	73	68
36696.375	3.322	51	2	10	65	58
36697.375	2.255	20	1	10	72	64
36698.375	2.146	20	2	10	75	72
36699.375	2.279	18	2	10	74	65
36703.375	1.204	5.4	1	10	76	70
36705.375	1.833	9.91	3	10	69	65
36706.375	2.111	6.25	2	10	64	61
36712.375	1.803	4.075	1	10	75	66
36713.375	1.911	5.8	2	10	72	64
36717.375	1.763	7.14	2	7	75	72
36718.375	2.322	13	3	6	73	68
36719.375	2.304	14	1	10	71	61
36720.375	1.544	13	1	10	72	65

Figure 11. Post-validation enabling of the Data Processing functionality

At this point, the grid cells (other than the ID column) are editable – that is, users can manually enter new numeric data into the cells by double-clicking on a cell and typing in a new value. $VB_{2.4}$ does not allow blank cells or non-numeric data in cells. Additionally, a right mouse-click on an IV column header presents options:

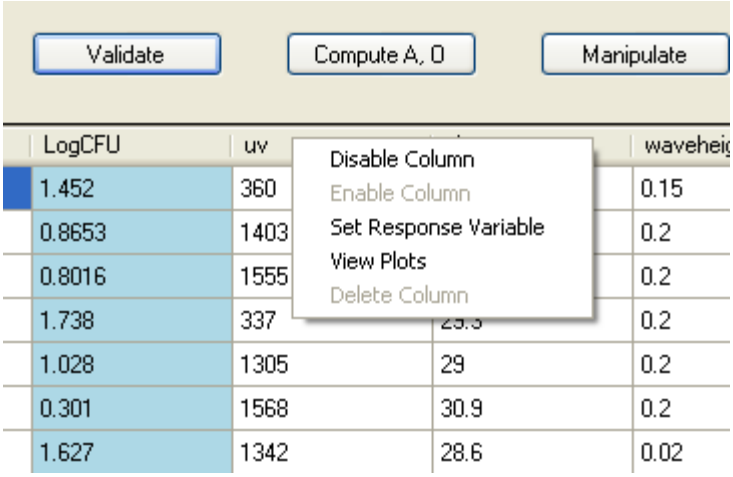


Figure 12. Right-click options on columns that are not the response variable

"Disable Column" turns the column's text red and prevents the column from being passed to the Modeling tab of $VB_{2.4}$. Previously-disabled columns can be activated using "Enable Column." "Set Response Variable" will make that IV the new response variable and it becomes blue as a visual indication of this change. "View Plots" shows a new screen with column statistics at the far left and four plots for that IV: (1) a scatterplot of the IV versus the response variable in the upper left panel, (2) a plot of the IV values versus the ID column at the upper right (a time series plot if the ID is an observation date), (3) a box-and-whiskers plot at the bottom left, and (4) a histogram for the IV at the bottom right.

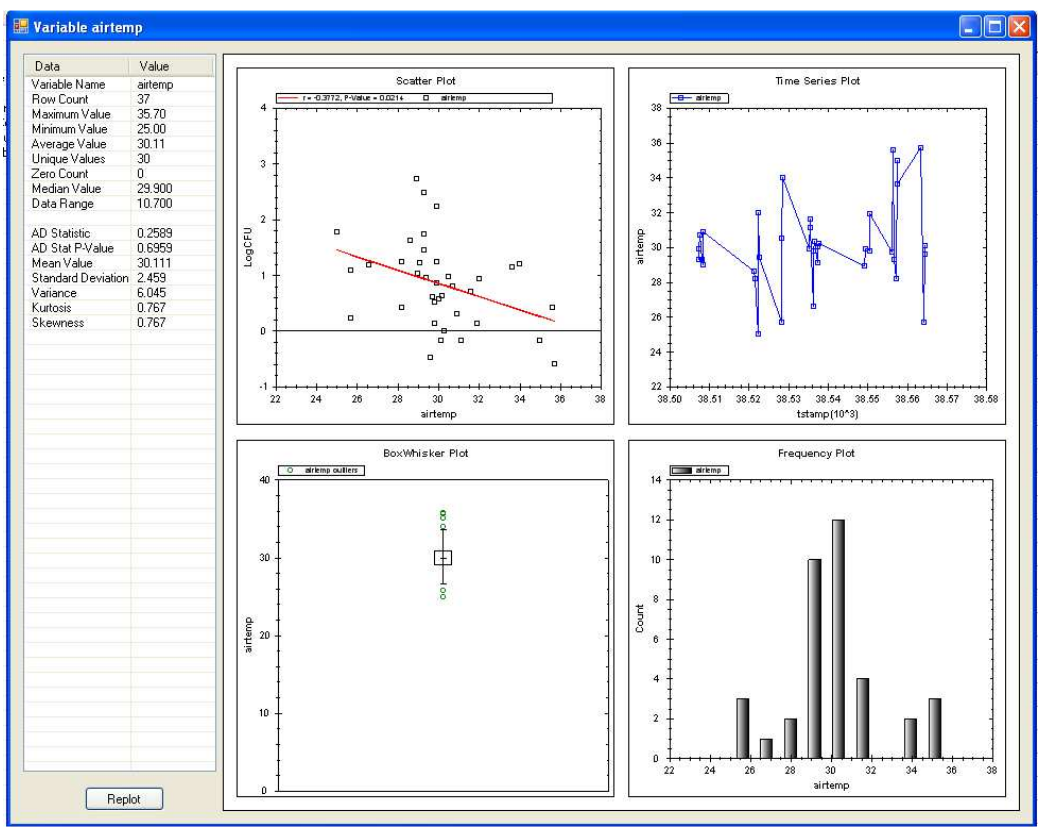


Figure 13. Four different plots available for evaluation of IVs

The scatter plot (upper left) is probably the most-examined, as it can indicate a non-linear relationship between the IV and the response variable, problems with homogeneity of variance across the range of the IV, or outliers. Ensuring that the IVs are linearly related to the response variable raises the probability of producing a robust, meaningful analysis. If the relationship between the response and the IV is not well-approximated by a straight line (a fundamental assumption of MLR), it may be beneficial to transform the IV. Using $VB_{2.4}$ to accomplish this will be explained later in this document. The scatterplot also shows the best-fit regression line in red, along with the correlation coefficient ("r") and the significance (p-value) of the correlation coefficient at the top of the plot. For the most part, p-values below 0.05 are considered statistically significant.

Identifying odd values (potential outliers or bad data) of any IV can often be done by visually inspecting these plots. If users double-click on the data point marker for any observation in one of the top panels or the bottom left panel (i.e., not the histogram), they can disable that point (the row) in the data grid.

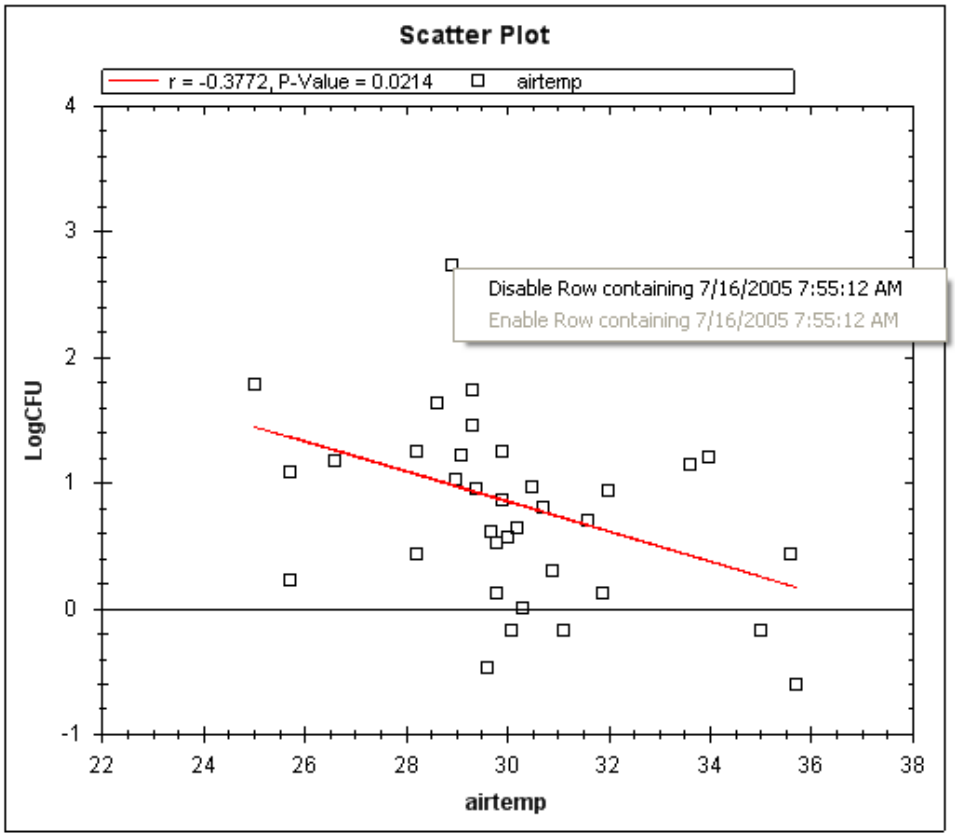


Figure 14. Disabling an observation from within the XY scatterplot

The final choice -- "Delete Column"-- deletes a column from the data grid, but the original columns of the imported data sheet ($VB_{2.4}$ defines these as "main effects") cannot be deleted. Rows can be disabled and enabled, but not deleted, from the data grid by right-clicking the row header (far left of each row) and making the desired choice.

If the user right-clicks on the column header of the response variable, a different set of choices is shown:

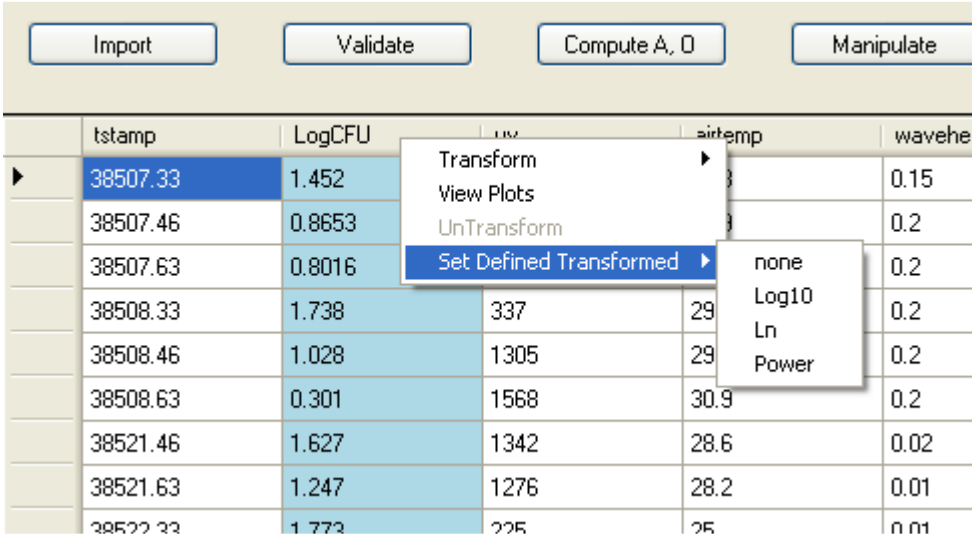


Figure 15. Available choices when right-clicking the current response variable

Users can transform the response variable in three ways: $\log_{10}$, $\log_e$, or a power transformation (raising the response to an exponent: y^{λ}). They can also un-transform the response, view the plots shown previously for the IVs, or define a transformation of the response variable. This option is used when a datasheet is imported with an already-transformed response variable. For example, users could import a datasheet with $\log_{10}$-transformed fecal indicator bacteria levels and then define the response as being $\log_{10}$-transformed. Doing this facilitates later comparisons with observations, decision criteria, and regulatory standards. When users transform the response variable within $VB_{2.4}$ using the "Transform" option, $VB_{2.4}$ automatically defines the response as having the chosen transformation and, in doing so, synchronizes the units of measurement for later comparisons.

6.5 Computing Alongshore and Onshore/Offshore Wind, Wave and Current Components

Orthogonal wind, current, and wave vectors can be powerful predictors of beach bacterial concentrations. Depending on the orientation of the beach, wind and currents can influence the movement of bacteria from a nearby source to the beach, and wave action can re-suspend bacteria buried in beach sediment. To make more sense of these data, researchers typically decompose wind/current/wave magnitude and direction into A (alongshore) and O (offshore/onshore) components for analysis (see equations at the end of this section).

If direction and magnitude (speed/height) data are available, A and O components can be calculated with the "Compute A, O" button. Clicking it brings up a window where users specify which columns of the data grid contain the relevant magnitude and direction data, using drop-down menus (Figure 16). There is also an input box at the bottom of the form for the beach orientation angle. If the user defined the angle on the "Beach Location" tab, that value should be seen here. After clicking "OK," new data columns are added to the far right of the data grid, representing the A and O components of the specified wind, current, or wave data. Unlike the originally imported IVs, these

components can be deleted from the data grid after they are created. Names of these new columns will be: WindA_comp(X,Y,Z), CurrentO_comp(X,Y,Z), WaveA_comp(X,Y,Z), etc, where X is the name of the column of data used for magnitude, Y is the name of the column used for direction, and Z is the beach orientation angle.

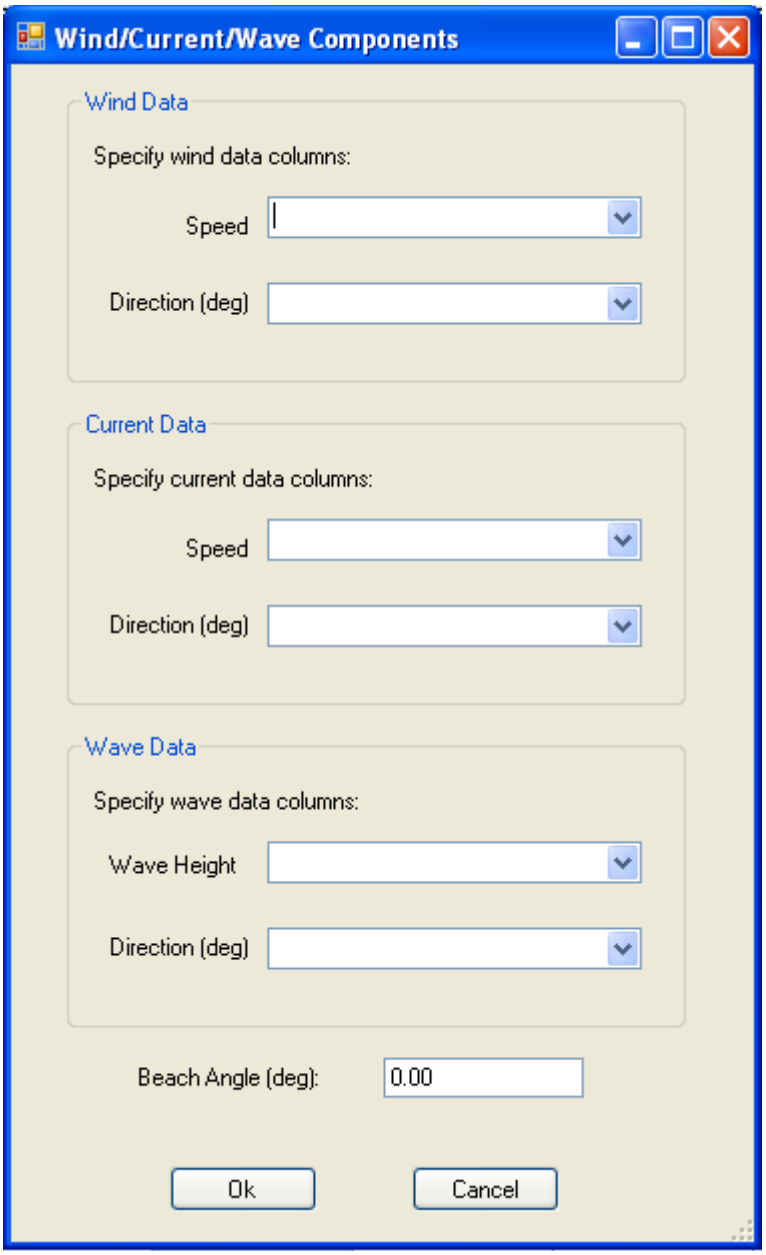


Figure 16. Window for computation of alongshore and offshore/onshore components

Notes on wind, wave and current component calculations:

Direction is an angular degree measure. Moving in a clockwise direction from north (0 degrees), values are positive, and negative while moving counter-clockwise. Wind and current speed (as well as wave height) can be measured in any unit. $VB_{2.4}$ adheres to scientific convention where wind direction is specified as the direction from which the wind blows, while current and wave directions are specified as the direction toward which the current or waves move. Thus, wind blowing from west to east has a direction

of 270 (or -90) degrees, while a current/wave moving from west to east has a direction of 90 degrees.

The A component measures the force of the wind/current/wave moving parallel to the shoreline (Figure 17). A positive A component means winds/currents/waves are moving from right to left as you look out at the water. A negative A component means winds/currents/waves are moving left to right as you look out at the water. The O component measures force perpendicular to the shoreline. A negative O value indicates movement from the land surface directly offshore (unlikely to see with wave action). A positive O indicates waves/wind/currents from the water to the shore. These relationships apply no matter how the beach is oriented (Figure 18).

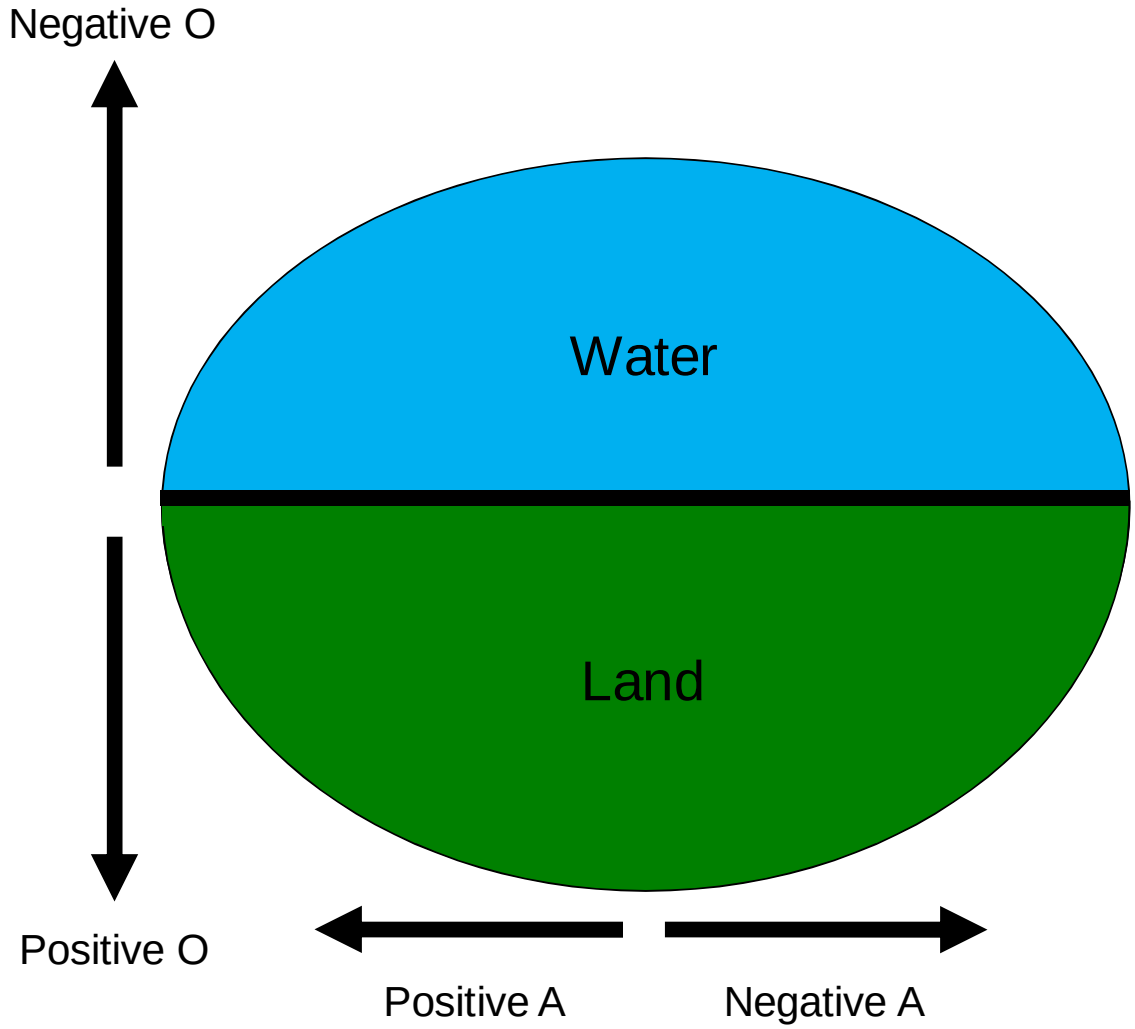


Figure 17. A and O component definitions for wind, current, and wave data

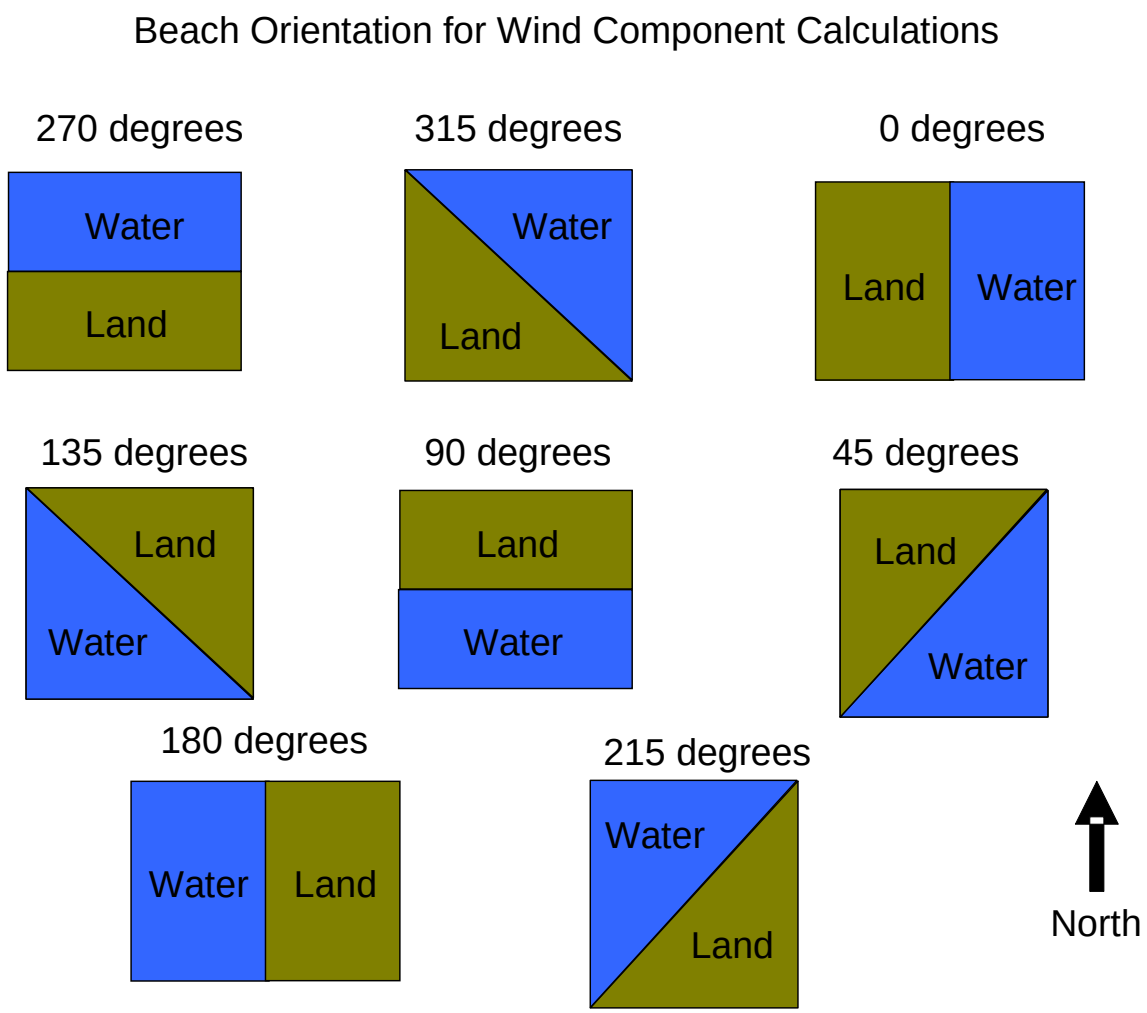


Figure 18. Principal beach orientations given in degrees

Equations for calculation of Wind A/O components:

Wind A: -SPD * cosine ((DIR-BO) * PI/180)
Wind O: SPD * sine ((DIR-BO) * PI/180)

where SPD is wind speed, DIR is wind direction, BO is the beach orientation (in degrees) and PI = 3.1416. Current A/O and Wave A/O are these same equations multiplied by -1.

6.6 Creation of New Independent Variables

Users may click the "Manipulate" button to create new columns of data that might serve as useful IVs. On the screen that pops up, there is a list of available IVs on the far left, under "Independent Variables." If users wish to create a new term, they add any available IV used in this new term by selecting it and using the ">" button to add it to the "Variables in Expression" box. Clicking and dragging down through the "Independent Variables" list allows for multiple IVs to be added at once.

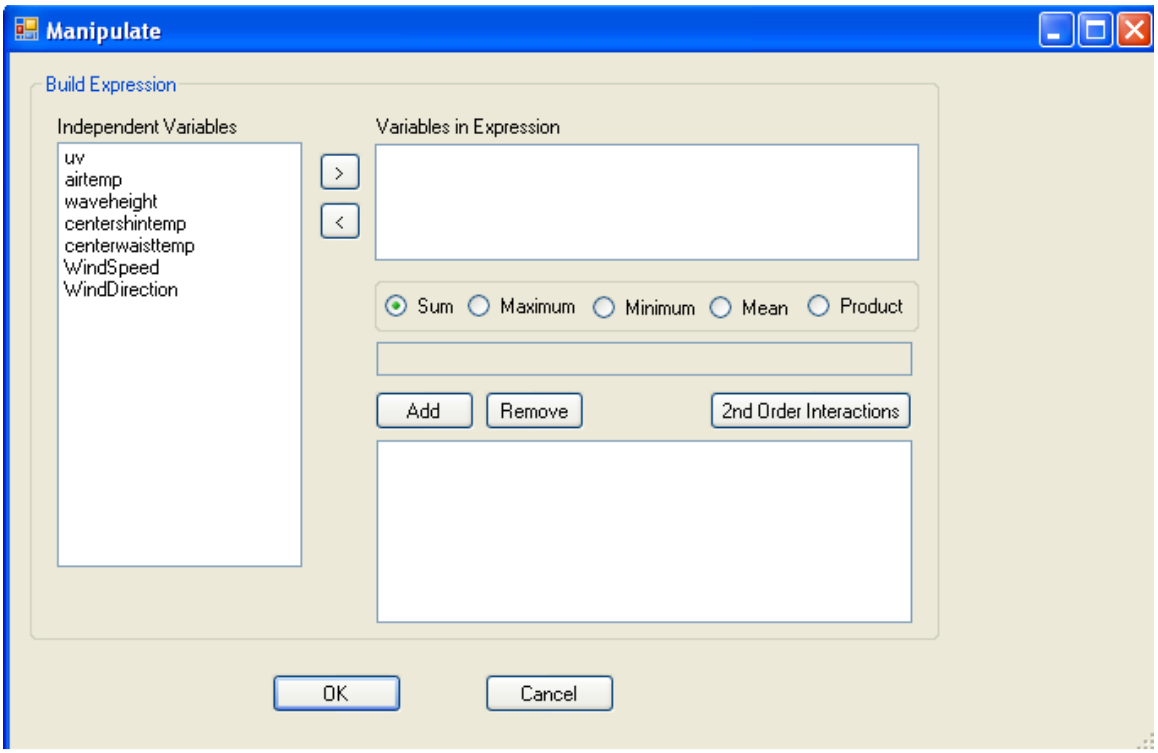


Figure 19. Window for the formulation of "Manipulates" - arithmetic combinations of existing columns within the data grid

For example, if users wish to create a new IV that is a row-by-row mean value of the "centershintemp" and "centerwaisttemp" variables, they add those two to the "Variables in Expression" box, then choose the "Mean" function, "Add" that expression to the lower box, then click "OK." That adds a new column of data that represents a row-by-row average of the two IVs, to the end of the data grid (far right.)

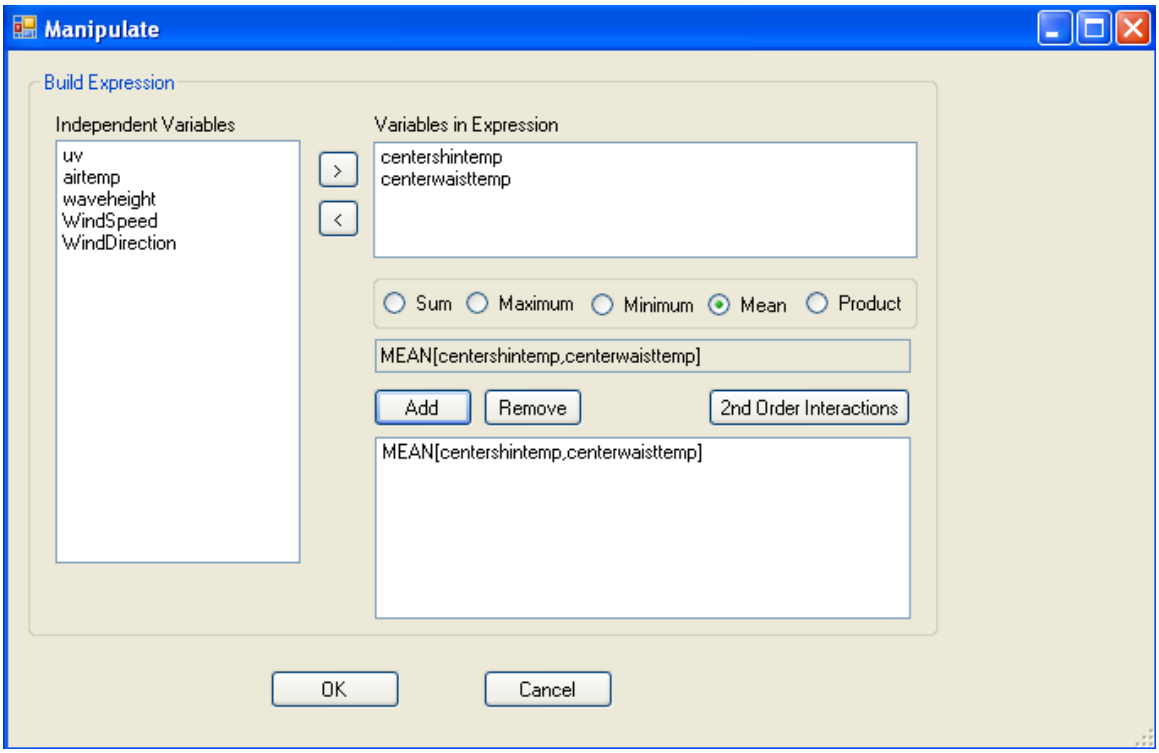


Figure 20. Creation of a new IV defined as the mean of two existent IVs

Users can create a row-by-row sum, maximum, minimum, mean, or product from any number of IVs that are added to the "Variables in Expression" box. More than one expression can be created before the "OK" button is clicked, and IVs can be easily moved in and out of the box using "<" and ">" keys. Any created expressions can be removed from the lower box with the "Remove" button. No matter how many IVs are added to the "Variables in Expression" box, clicking "2nd Order Interactions" will add the cross-products for all possible pairings of those IVs. Thus, four IVs will produce six interactions, five IVs will produce ten interactions, and so on. Note that the names of the columns used to create any manipulate are inside the parentheses of that manipulate's column name.

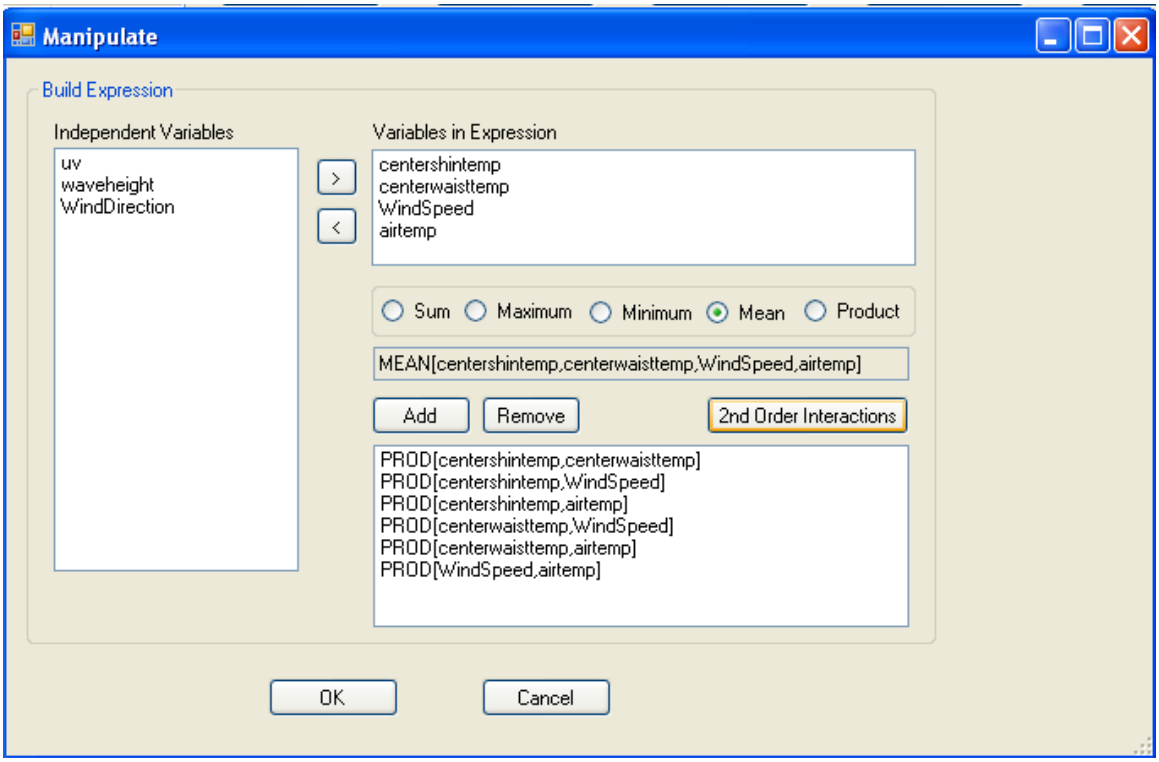


Figure 21. Formation of two-way cross-products of a set of four existent IVs

$VB_{2.4}$ does not allow previously created "manipulates" -- new columns of data created through the "Manipulate" button -- to be further manipulated. Previously-created manipulates will not appear in the "Independent Variables" section at the left. They can, however, be chosen as the response variable or deleted from the data grid, using the appropriate menu choices, accessed by a right-click of the column header.

6.7 Transforming the Independent Variables

$VB_{2.4}$ gives users the ability to transform non-categorical IVs to assist in linearizing the relationship between the IVs and the response variable, which is a fundamental assumption of an MLR analysis. $VB_{2.4}$ provides the following transformations, where X_t is the transformed IV and X is the original IV:

Log_{10}: $X_t = \log_{10}(X)$
Log_e: $X_t = \log_e(X)$
Inverse: $X_t = 1/X$
Square: $X_t = X^2$
Square Root: $X_t = X^{0.5}$
Quad Root: $X_t = X^{0.25}$
Polynomial: $X_t = a + bX + cX^2$
General Exponent: $X_t = X^e$ where the user specifies the value of e

When users click the "Transform" button, they are presented a choice of transformations to investigate:

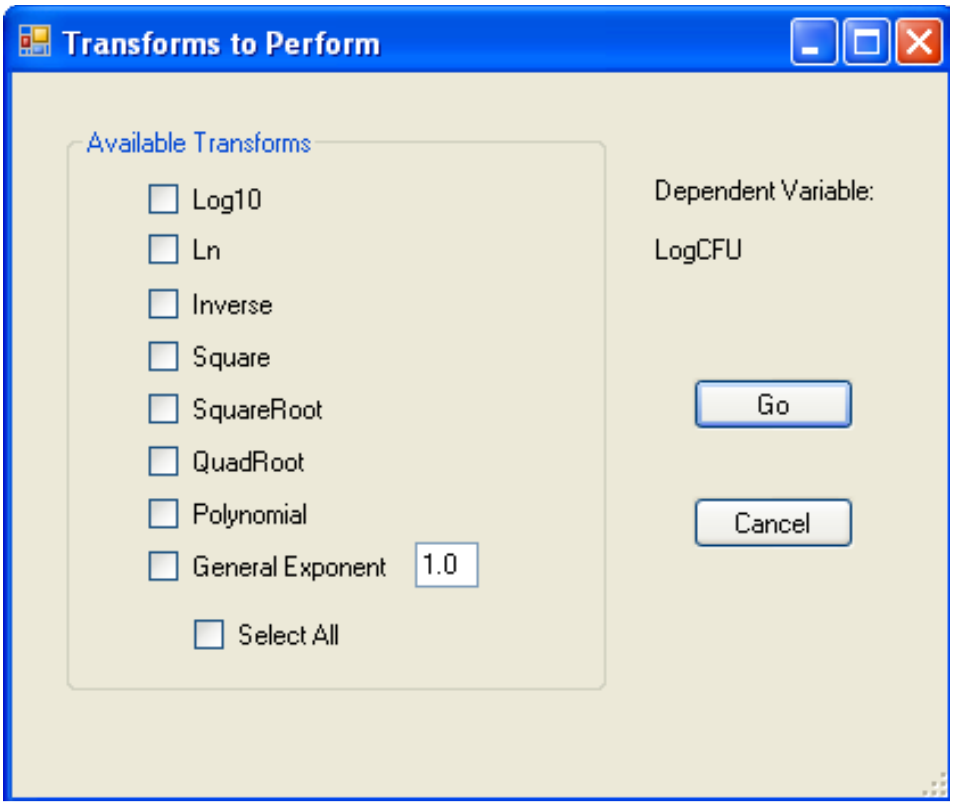


Figure 22. The range of choices for IV transformations

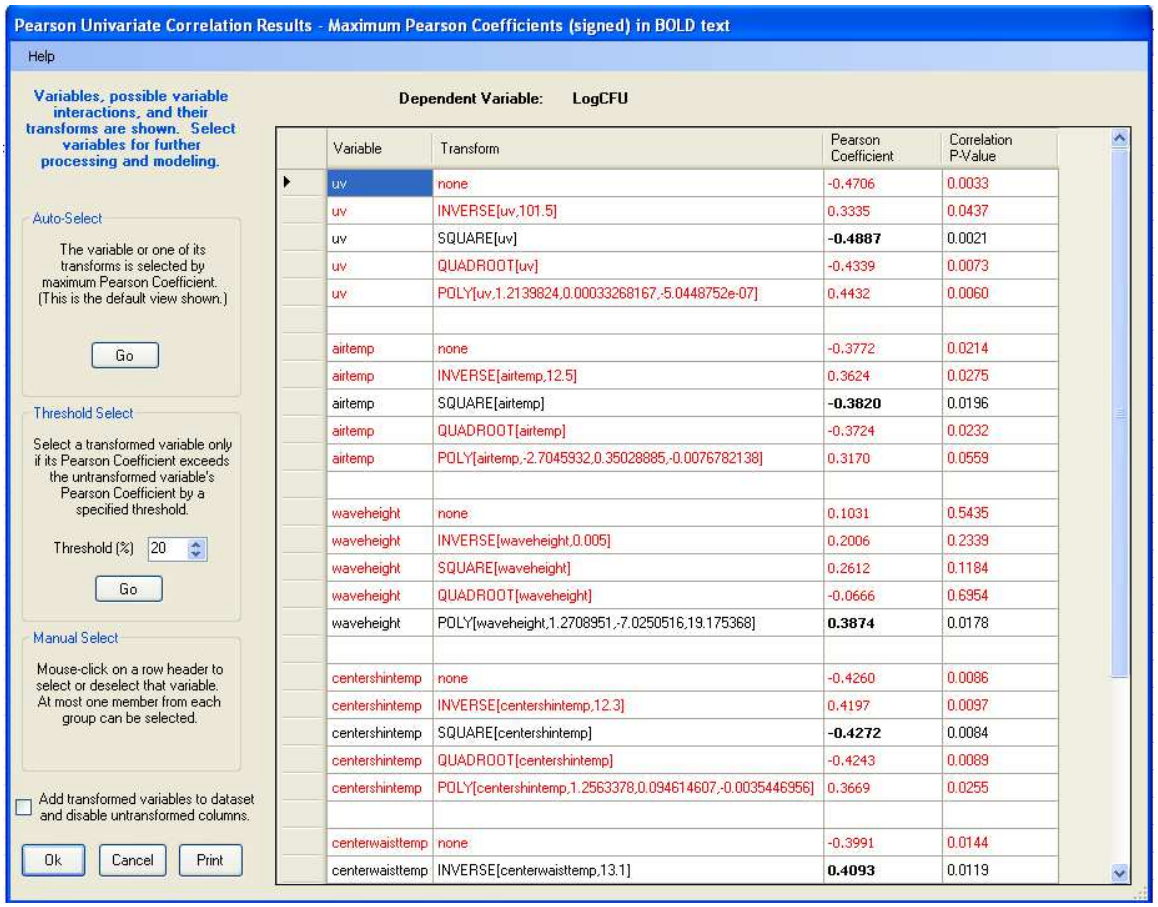
When users click "Go", the chosen transforms are applied to each non-categorical IV. $VB_{2.4}$ then opens a table that allows comparison of the success of each transform using a Pearson correlation coefficient, a measure of linear dependence between the response variable and the IVs. For the polynomial transformation, the Pearson coefficient is calculated as the square root of the adjusted R^2 value derived from the regression of the response on X_t. Because this adjusted R^2 value can possibly be negative, an empirically-derived formula is applied when adjusted R^2 values fall below 0.1:

$$\text{Polynomial Pearson Coefficient} = (-6.67*RE_1^2 + 13.9*RE_1 - 6.24)*(R^2)^{0.5}$$

where $RE_1 = 1.015 - 1.856*R^2 + 1.862*adjR^2 - 0.000153*N$, R^2 and $adjR^2$ are defined by the regression of the response on X_t, and N = number of observations.

The table that $VB_{2.4}$ creates groups all transformed versions of each IV by the IV name, type of transformation, and the associated Pearson coefficient. By default, the transformation (this includes the un-transformed version of the IV, denoted by "none"), with the largest absolute value of the Pearson coefficient is highlighted in black text for selection. Users may override the default selection by left-clicking on the row header of a transformed IV they choose. They may also override the default by setting a Threshold percentage and clicking "Threshold Select" on the left side of the box. This selects the un-transformed IV unless the transformed IV with the highest absolute value Pearson coefficient exceeds the un-transformed IV Pearson coefficient by the specified percentage. In essence, the user is saying, "Unless the Pearson coefficient of the transformed IV is some % greater than the Pearson coefficient of the un-transformed IV,

use the un-transformed IV." This can be useful because transforming IVs makes interpreting model coefficients more difficult; unless an improvement is seen, transformation may not be worth the trouble. Users can also revert to the default by clicking "Go" under the "Auto Select" section at the left.



Variable	Transform	Pearson Coefficient	Correlation P-Value
uv	none	-0.4706	0.0033
uv	INVERSE[uv,101.5]	0.3335	0.0437
uv	SQUARE[uv]	**-0.4887**	0.0021
uv	QUADROOT[uv]	-0.4339	0.0073
uv	POLY[uv,1.2139824,0.00033268167,-5.0448752e-07]	0.4432	0.0060
airtemp	none	-0.3772	0.0214
airtemp	INVERSE[airtemp,12.5]	0.3624	0.0275
airtemp	SQUARE[airtemp]	**-0.3820**	0.0196
airtemp	QUADROOT[airtemp]	-0.3724	0.0232
airtemp	POLY[airtemp,-2.7045932,0.35028885,-0.0076782138]	0.3170	0.0559
waveheight	none	0.1031	0.5435
waveheight	INVERSE[waveheight,0.005]	0.2006	0.2339
waveheight	SQUARE[waveheight]	0.2612	0.1184
waveheight	QUADROOT[waveheight]	-0.0666	0.6954
waveheight	POLY[waveheight,1.2708951,-7.0250516,19.175368]	**0.3874**	0.0178
centershintemp	none	-0.4260	0.0086
centershintemp	INVERSE[centershintemp,12.3]	0.4197	0.0097
centershintemp	SQUARE[centershintemp]	**-0.4272**	0.0084
centershintemp	QUADROOT[centershintemp]	-0.4243	0.0089
centershintemp	POLY[centershintemp,1.2563378,0.094614607,-0.0035446956]	0.3669	0.0255
centerwaisttemp	none	-0.3991	0.0144
centerwaisttemp	INVERSE[centerwaisttemp,13.1]	**0.4093**	0.0119

Figure 23. Pearson correlation coefficient scores for judging the efficacy of IV transformations

Plotting Transformed IVs

Users may prefer to examine plots visually to determine which transformation of IV to choose. If users right-clicks on a row header in this correlation table, they can view an array of scatterplots, time series plots, or frequency plots for each data transformation of the IV represented by that header. Scatterplots will show the best-fit regression line, the correlation coefficient, and the p-value for that correlation coefficient.

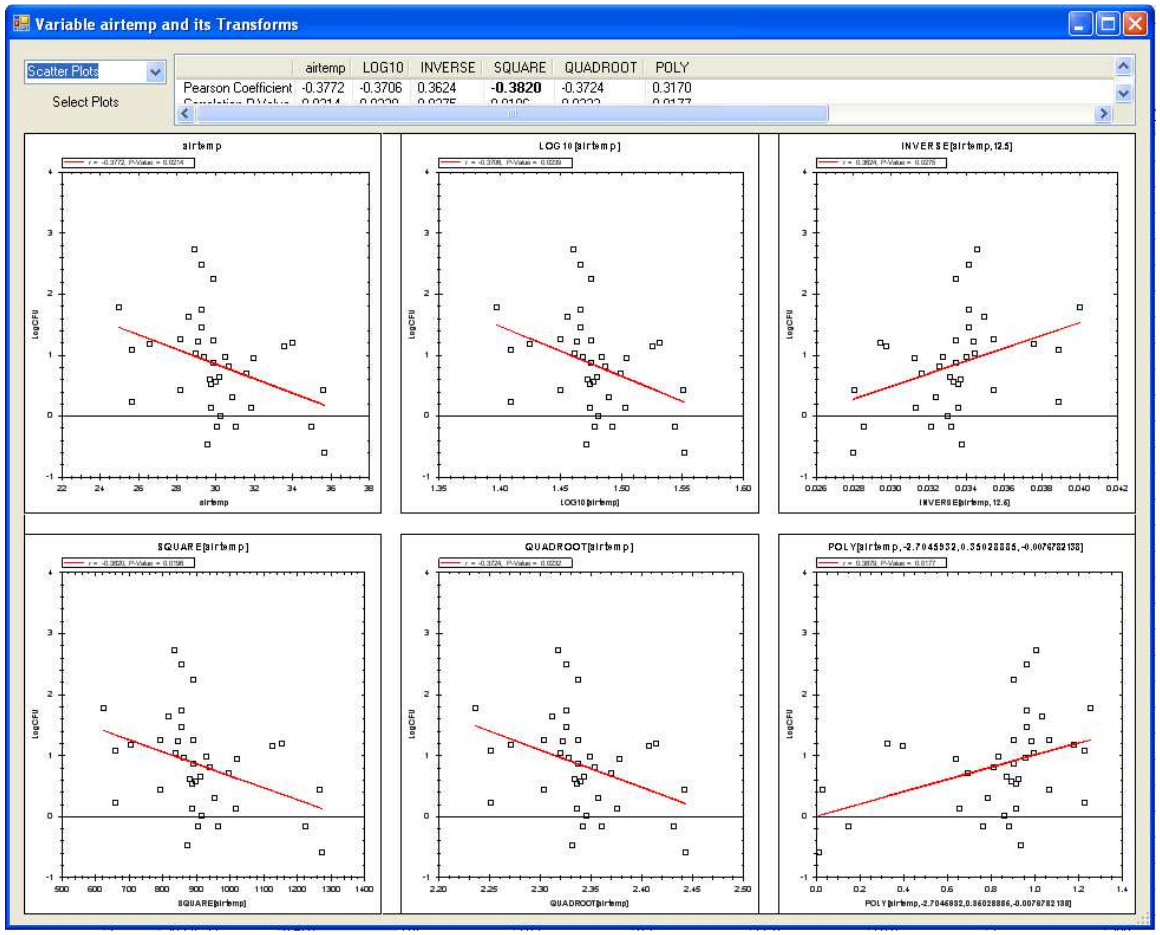


Figure 24. Scatterplots (Response vs. IV) for six different data transformations of a single IV

After choosing a transformation for each IV, users click "OK." This populates the data grid with new columns representing transformed versions of the IVs. The small checkbox in the bottom left corner (Figure 23) controls whether the untransformed version of the IV remains enabled in the data grid after the user clicks "OK." When the box is checked, for any IV in which the user chooses a transformed version, the un-transformed version will be disabled in the data grid. Notice that transformed versions of an IV are put into the data grid immediately after the original, un-transformed IV.

Notes on Transformed IVs

Any transformations put into the data grid can be deleted with the "Delete Column" choice after right-clicking on their column header. Transformed IVs will appear in the list of IVs on the "Manipulate" screen; however, transformed IVs cannot be

further transformed and will not appear in the transform table if the user goes back to the "Transform" window.

$VB_{2.4}$ transformations have specific processing for certain data values and are not pure mathematical transformations -- they were designed to maintain data order while helping to linearize the response-IV relationship. For the SQUARE (b=2), SQUAREROOT (b=0.5), QUADROOT (b=0.25), INVERSE (b=-1) and GENERAL EXPONENT (b is user-defined) transformations, $VB_{2.4}$ uses the signed equivalent of the mathematical function:

$$x\text{^}b == \text{sign}(x)*\text{abs}(x)\text{^}b$$

For example: $(-2)^2 = -4$ $(-9)^{0.5} = -3$ $(-4)^{-0.5} = -0.5$ $(-2)^{-2} = -0.25$

To avoid potentially undefined values (i.e., 1/x when x = 0), the INVERSE and GENERAL EXPONENT (if the user sets b < 0) transformations have special processing:

If x = 0, then $VB_{2.4}$ will find the minimum of abs(z), where z is the set of all non-zero values for the IV in question. For the purpose of computing the transformation, once z is defined, $VB_{2.4}$ substitutes z/2 for x. From this definition, note that z can be either a positive or negative number.

LOG_{10} and LOG_e transforms are also the signed equivalent of the mathematical functions:

$$\log_e(x) == \log_e(x)$$
$$\log_e(-x) == -\log_e(x)$$
$$\log_{10}(x) == \log_{10}(x)$$
$$\log_{10}(-x) == -\log_{10}(x)$$

In addition, if $(-1 \leq x \leq 1)$, then $\log_e(x) = 0$ and $\log_{10}(x) = 0$

$VB_{2.4}$ will not compute the INVERSE, GENERAL EXPONENT (with a negative b), LOG_{10} and LOG_e transformations for data columns if more than 10% of the IV's values are zero. Programmatically, zero is defined as any number whose absolute value is less than 1.0e-21.

POLYNOMIAL transformations are the result of a linear regression of the response variable on the IV and the square of the IV:

$$\text{Poly}(X) = a + b*X + c*X^2$$

where a, b, and c are determined by a multiple linear regression of X and X^2 on the response variable.

In general, the name of the transformed column of data that $VB_{2.4}$ creates is simply the type of transformation, with the original data column name in parentheses. For example, WaterTemp would become LOG_{10}(WaterTemp). There are some exceptions, however:

INVERSE(X,Y) : X is the original data column name and Y is the z/2 value discussed earlier in this section.

POWER(X,Y) : When Y is positive, X is the original data column name and Y is the exponent specified by the user.

POWER(X,Y,Z) : When Y is negative, X is the original data column name, Y is the exponent specified by the user, and Z is the z/2 value discussed earlier in this section.

POLY(X, a,b,c) : X is the original data column name and a, b, and c are the values of the polynomial regression coefficients.

Finally, because transformations are determined by the current response variable, when users change the response variable in the data grid (using the column header right-click menu), all transformed IVs in the data grid are erased (a message warns the user).

6.8 Saving Processed Data

Data can be saved in a project file (Project→Save) at any time during data processing. When the file is opened, the data grid will be repopulated as it appeared when the project was saved. Also, users may highlight the entire table or sections of the table and use Control-c and Control-v to copy and paste the data grid into a word processing or spreadsheet application.

6.9 Go to Modeling

After data processing is complete, users must click the “Go to Modeling” button to open the Modeling tab. If users have already done modeling work and returned to the data sheet to make changes, they will receive a message that the data sheet has changed and any prior efforts on the Modeling and Prediction tabs will be erased. Users can then choose to move forward to the Modeling tab or return to the data sheet.

7. MODELING

The Modeling tab facilitates finding the best model based on criteria selected by the user. As the number of IVs increases, the number of possible models in the solution space increases exponentially. Users may select all or a subset of the IVs for consideration in the model to reduce the size of the solution space.

7.1 Selecting Variables for Model Building

All eligible IVs are listed in the left column ("Available Variables") under the Variable Selection sub-tab. Any variable users wish to consider for model inclusion must then be moved to the "Independent Variables" list by highlighting the IV and clicking the ">" key. Any number of IVs can be moved or removed from this list.

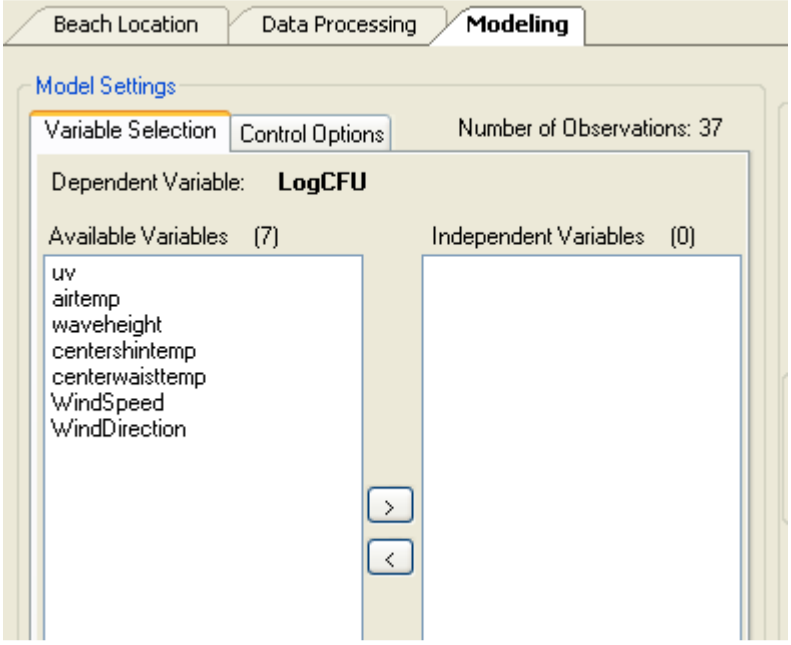


Figure 25. Selecting variables for MLR processing within the Modeling tab

As you add or remove IVs from the "Independent Variables" list, the number of possible MLR models is displayed in the status strip at the bottom right of the application window. The number of possible models can grow exceedingly large; 66 IVs represent $7.38*10^{19}$ possibilities. More than 66 variables produces a number that exceeds the capacity of the program to store it – in such cases, "more than 9.2e019" is displayed.

7.2 Modeling Control Options

The first decision users make on this tab involves which evaluation criteria will be used to judge model fitness. There are currently ten criteria available in the drop-down menu:

- Akaike Information Criterion (AIC)
- Corrected Akaike Information Criterion (AICC)
- R^2
- Adjusted R^2
- Predicted Error Sum of Squares (PRESS)
- Bayesian Information Criterion (BIC)
- RMSE
- Sensitivity
- Specificity
- Accuracy

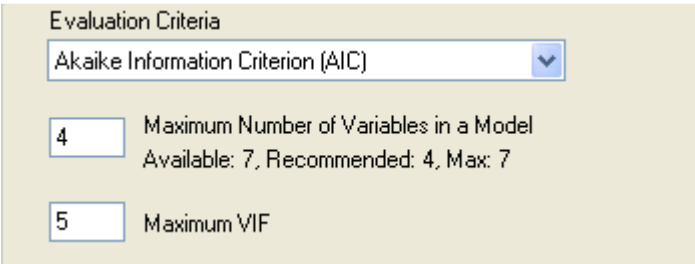


Figure 26. Setting modeling options within the Modeling interface

The "Maximum VIF" (Variance Inflation Factor) parameter is used selectively to discard models that contain variables with a high degree of multi-colinearity, i.e., IVs that are greatly correlated with other IVs. If any IV in a model has a VIF exceeding the threshold, that model will be discarded. The default VIF value used in the application is set to 5. A VIF of 5 means that 80% (1/5) of the variability in an IV can be explained by the variability of other IVs in the model. A VIF of 10 means that 90% (1/10) of the variability can be explained, and so on. If users aren't concerned with muli-collinearity among the explanatory variables in a regression model, they can lower the Maximum VIF value. However, multi-colinearity leads to poorly estimated regression coefficients (i.e., large standard deviations of these coefficients).

The "Maximum Number of Variables in a Model" parameter tells $VB_{2.4}$ how large the models being evaluated can be. As a rule, most modelers prefer to have about 10 observations per estimated parameter in their models, otherwise possibilities increase for model over-fitting and poor estimation of regression parameters. $VB_{2.4}$'s recommendation is close to this rule. It equals (1 + n/10) where n is the number of observations in the dataset. The maximum allowable number equals n/5. $VB_{2.4}$ won't let users set this value over the maximum. The total number of available parameters is also given here.

If we define *p* as the number of parameters in a model, *n* as the number of observations in the dataset, RSS as the residual sum of squares for a model, and TSS as the total sum of squares for a model, then the evaluation criteria for a model can be defined as:

● Akaike Information Criterion (AIC): $2p + n*\ln(RSS)$

● Corrected Akaike Information Criterion (AICC): $\ln(RSS/n) + (n+p)/(n-p-2)$

● R^2: $1 - RSS/TSS$

- Adjusted R^2: $1 - (1-R^2)(n-1)/(n-p-1)$

- Bayes (Schwarz) Information Criterion (BIC): $= n*\ln(RSS/n) + p*\ln(n)$

- Root Mean Squared Error (RMSE): $(RSS/n)^{1/2}$

- Predicted Error Sum of Squares (PRESS): $1 - \Sigma(y_i - \hat{y}_{-i})^2 / \Sigma(y_i - y_m)^2$
where y_i is the i_{th} observation, $\hat{y}_{-i}$ is the model estimate of the i_{th} observation when the model coefficients are fitted with the i_{th} observation removed from the dataset, and y_m is the mean value of y in the dataset

- Accuracy: (true positives + true negatives) / number of total observations

- Specificity: true negatives / (true negatives + false positives)

- Sensitivity: true positives / (true positives + false negatives)

Sensitivity, specificity and accuracy are special cases that require users to enter both a Decision Criterion (DC) and Regulatory Standard (RS) so that true/false positives and true/false negatives can be defined. The DC is a modeled (predicted) value the user chooses. Model predictions above this threshold are considered exceedances/positives, while model predictions below this value are considered non-exceedances/negatives. The RS is typically a safety limit on fecal indicator bacteria (FIB) levels set by a state or federal agency. The "Threshold Transform" radio buttons tell $VB_{2.4}$ how to transform the DC and RS for comparison to model predictions and observations. If a transformation definition is set for the response variable (either manually by the user or automatically by transforming the response) during data processing, that definition will be set as the default here. Users should understand that changing the threshold transform definition can lead to problems when comparing modeling predictions to observations. Caution should be exercised.

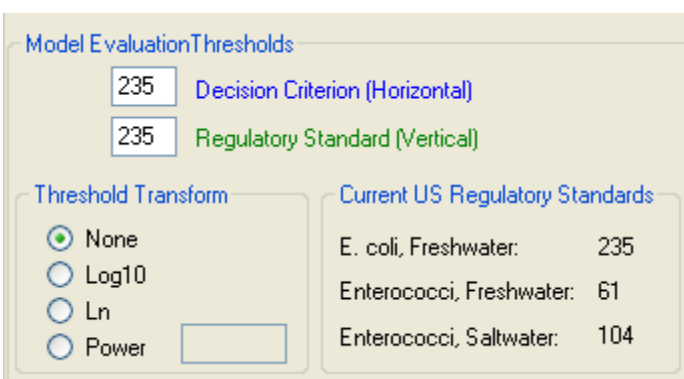


Figure 27. Setting evaluation thresholds and threshold transformation information within the modeling interface

7.3 Linear Regression Modeling Methods

There are two options for exploring the solution space.

1. Manual – this option is for a directed model search. If the 'Run all combinations' box is not checked, a single model including every IV that was added to the "Independent Variables" column will be evaluated. If 'Run all combinations' is checked, an exhaustive search is performed. The exhaustive search evaluates every model that can be constructed with the selected IVs, but does not evaluate any with more parameters than the "Maximum Number of Variables in a Model" input box. For example, if there are 24 IVs to evaluate and the maximum number of IVs in a model is set at 8, the exhaustive routine examines every possible 1, 2, 3, 4, 5, 6, 7 and 8-parameter model. As the number of IVs rises, the number of possible models quickly gets so large that the exhaustive routine cannot maintain reasonable computation times and the user is advised to switch to the genetic algorithm.

2. Genetic Algorithm – the Genetic Algorithm (GA) option explores solution spaces too large to handle exhaustively. Genetic algorithms are loosely based on the natural evolutionary process, in which individuals in a population reproduce and mutate. Individuals with high fitness (regression models that produce small residuals) are more likely to reproduce and pass their genes (IVs) to the next generation. The goal is to find a good solution without having to examine every possible option and the GA balances random and directed searching.

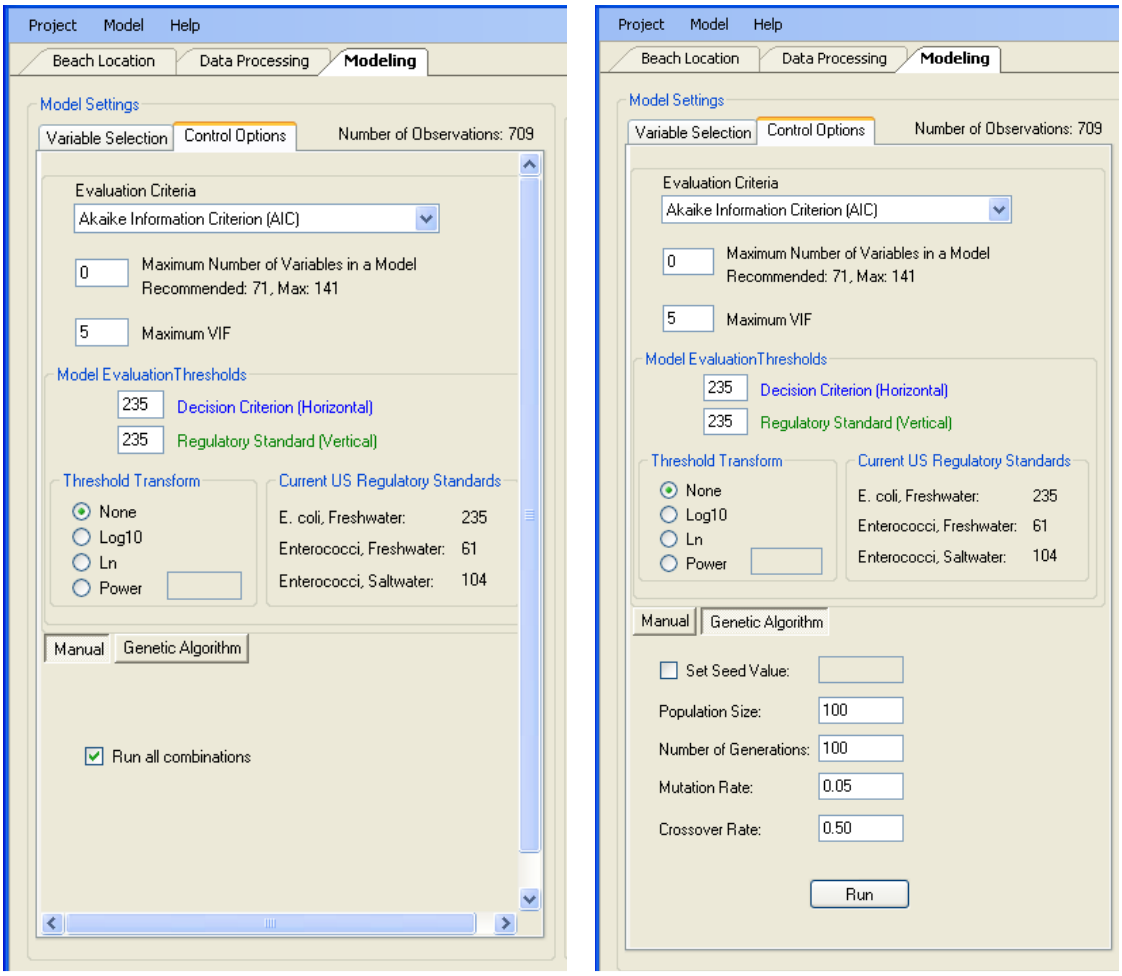


Figure 28. Model building interface using a manual search (left panel) or the Genetic Algorithm (right panel)

Choosing between an exhaustive and a GA search depends on your data set, available hardware and time constraints. Fifteen IVs produce about 32,000 model possibilities; on our system (Dell Precision T5400 workstation running MS Win XPSP3 w/ dual Xeon 2.66 GHz processors having 4 GB RAM), the exhaustive search was completed in approximately 90 seconds. Sixteen IVs represent more than 65,000 possibilities which is more than double that of 15 IVs. Some model building results are summarized below:

Exhaustive Search – Run All Combinations		
Number of IVs	Number of MLR models	Approximate Time Required to Generate and Filter Models (seconds)
15	32767	90
16	65535	110
17	131071	280

By contrast, the GA with 17 IVs was completed in less than seven seconds. We note, however, that the exhaustive search did find a slightly better model than the GA did using the selected AIC evaluation criterion (49.2 versus 55).

An alternative modeling strategy could be to use the GA on your entire list of IVs, then the exhaustive search on a subset of the initial IVs – any IV that appears in one of the best ten models found by the GA. This two-step modeling process is facilitated with the "IV Filter" list control.

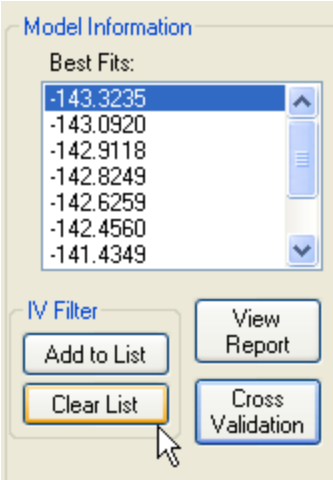


Figure 29. Using the IV filter to select a subset of variables from the best-fit models

When the GA ends and the 10 best models are shown, use the "Clear List" button to remove all IVs from the selection list. Select a model from the "Best Fits" list one at a time and click the "Add to List" button; this action adds any IVs in the model to the Independent Variable list. After doing this for the ten best models, users likely have a much more manageable IV list and can run an exhaustive search to find the very best combination of IVs. Regardless of the method chosen to build models, the "Best Fits" window shows the top ten models found, in terms of the evaluation criterion chosen.

7.4 Using the Genetic Algorithm

There are five parameters users can set to adjust performance of the GA:

a) Seed value: internal random number generator to produce random values. Setting this seed to a known value will make the GA run reproducible. Changing the seed will create a new series of random values, possibly returning different results.
b) Population size: number of individuals in the population of each generation. A larger population broadens the search at each generation, but slows processing time.
c) Number of generations: how long to run the search since individuals can reproduce and mutate once each generation. The fitness of every individual in the population is evaluated at the end of each generation.
d) Mutation rate: chance each individual has of undergoing random mutation in each generation. The higher the mutation rate, the more random (less directed) the search of parameter space is.

e) Crossover rate: probability that two selected individuals in the population will exchange genome parts. Exchanging genes creates new individuals in the population.

The best GA parameter values depend on the dataset being investigated, but typical values of the mutation rate are between 0.001 and 0.1 (0.1 and 10%) and typical values of the crossover rate are between 0.4 and 0.75 (40 and 75%). For most datasets, a population size and generation number of 100 will be sufficient. Larger datasets may require an increase in these numbers for optimal solutions.

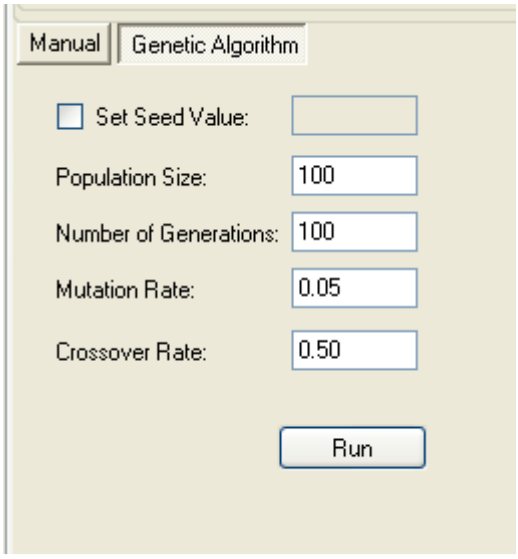


Figure 30. Genetic algorithm options within the modeling interface

7.5 Evaluating Model Output

After selecting a method to build models and an evaluation criterion to rank them, users then click the "Run" button. Model selection and evaluation progress is displayed on the "Progress" graph at the lower right of the Modeling tab. Note that the "Run" button changes to "Cancel;" the process is interruptible should progress be unacceptably slow. Once model-building is completed, the ten best MLR fits are displayed in the "Best Fits" box. Selecting a model from the list results in (see Figure 31):

1. A list of the selected model's IVs with associated regression coefficients and statistics is displayed on the "Variable Statistics" subtab.
2. A list of the selected model's evaluation metrics is shown on the "Model Statistics" subtab.
3. The "Results" subtab will show two data series - the model fits and observations – versus the observation number. If observations are chronologically ordered, this is basically a time series plot of the two data series.
4. The "Fitted vs Observed" subtab shows plots and tables based on model fitted values versus the observations.

5. The “ROC Curves” subtab shows a plot of the Receiver Operating Characteristic curve of each “Best Fits” model, as well as a table showing the computed AUC (area-under-the-curve) for each ROC (see Section 7.7).
6. Clicking on “View Report” generates a text report of model and variable statistics for the selected model.
7. The “Residuals” subtab allows users to access the residual analysis functions of the application (see Section 7.8).
8. The “MLR Prediction” tab will appear at the top, allowing users to proceed to the prediction component of the application.

Note that selecting a different model from the “Best Fits” list updates the Variable and Model Statistics tables, as well as the information displayed on the “Results,” “Fitted vs Observed,” “ROC Curves” and “Residuals” subtabs.

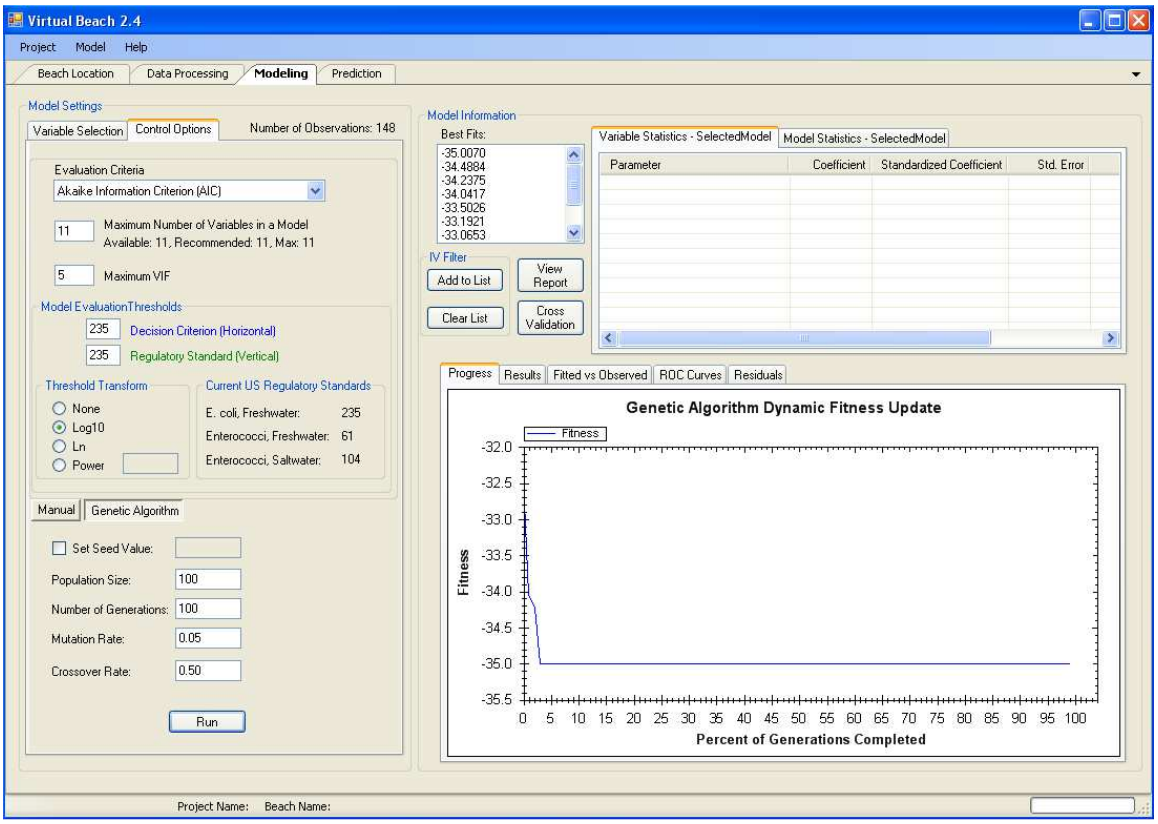
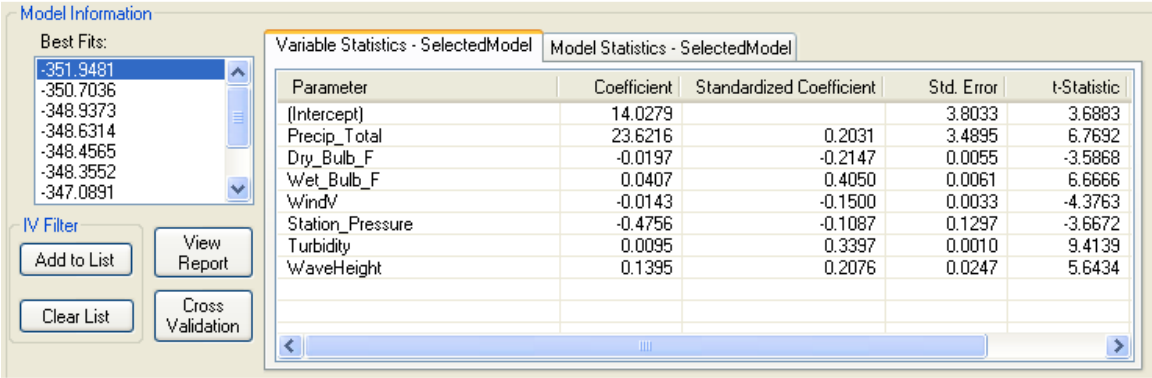
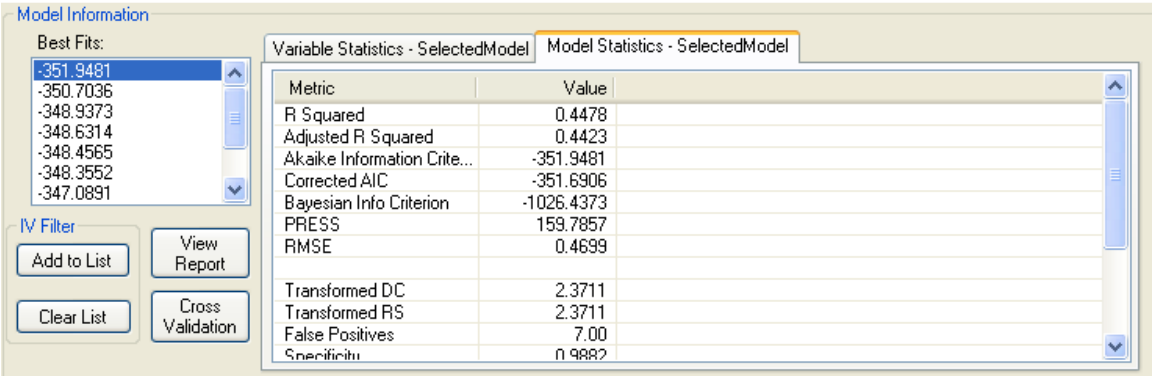


Figure 31. Modeling results shown after completion of a run using the genetic algorithm



Parameter	Coefficient	Standardized Coefficient	Std. Error	t-Statistic
(Intercept)	14.0279		3.8033	3.6883
Precip_Total	23.6216	0.2031	3.4895	6.7692
Dry_Bulb_F	-0.0197	-0.2147	0.0055	-3.5868
Wet_Bulb_F	0.0407	0.4050	0.0061	6.6666
WindV	-0.0143	-0.1500	0.0033	-4.3763
Station_Pressure	-0.4756	-0.1087	0.1297	-3.6672
Turbidity	0.0095	0.3397	0.0010	9.4139
WaveHeight	0.1395	0.2076	0.0247	5.6434

Figure 32. Modeling Interface showing variable statistics for the selected Best-Fit model



Metric	Value
R Squared	0.4478
Adjusted R Squared	0.4423
Akaike Information Crite...	-351.9481
Corrected AIC	-351.6906
Bayesian Info Criterion	-1026.4373
PRESS	159.7857
RMSE	0.4699
Transformed DC	2.3711
Transformed RS	2.3711
False Positives	7.00
Specificity	0.9882

Figure 33. Modeling interface showing model evaluation metrics for the selected Best-Fit model

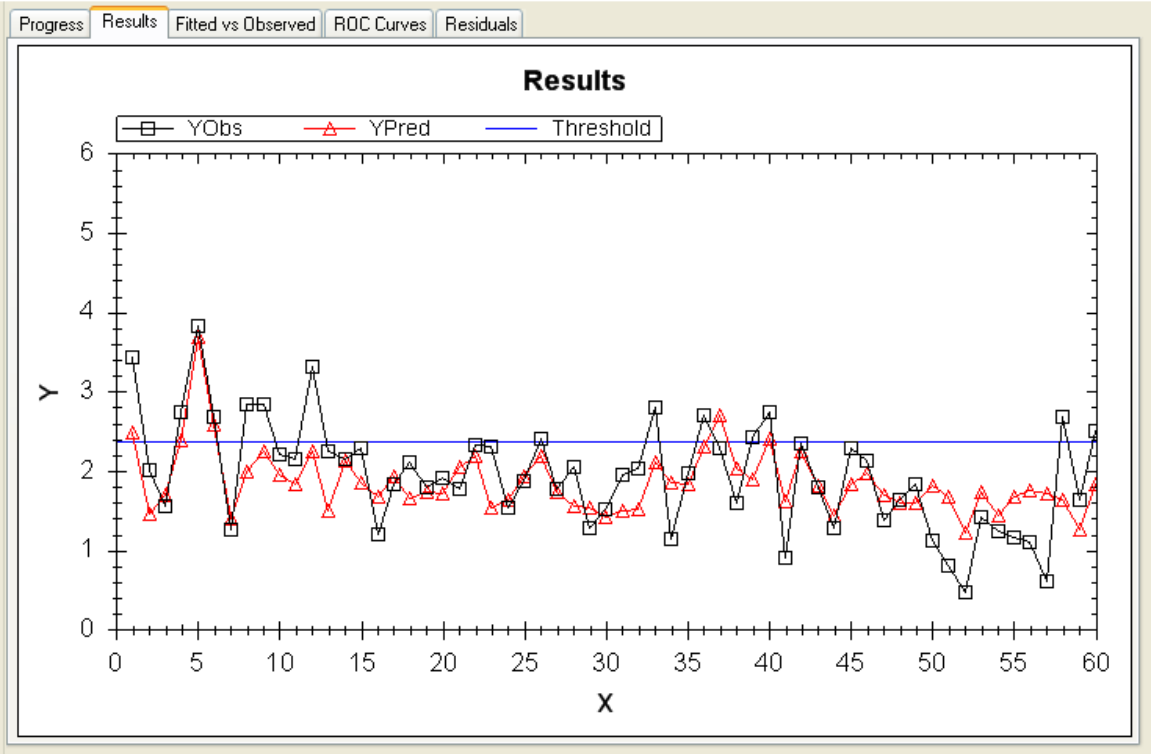


Figure 34. Modeling interface showing a time series plot for the selected model

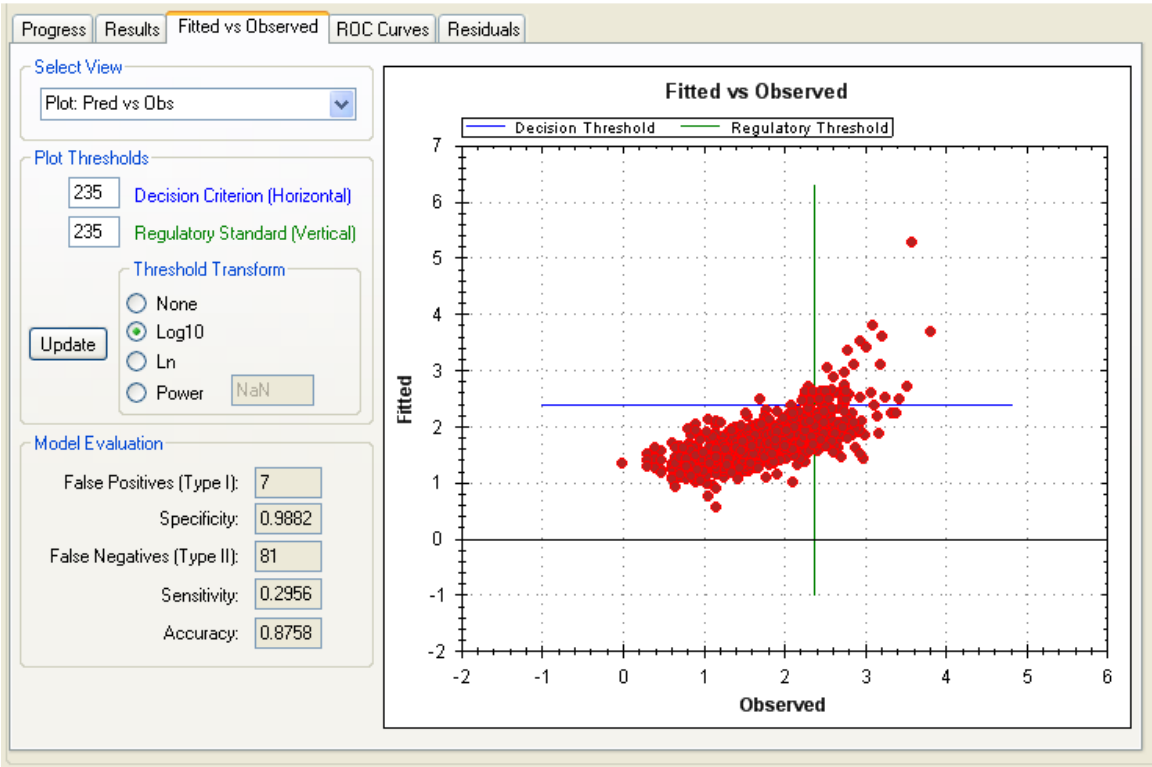


Figure 35. A scatter plot of fitted values of the selected model versus observations

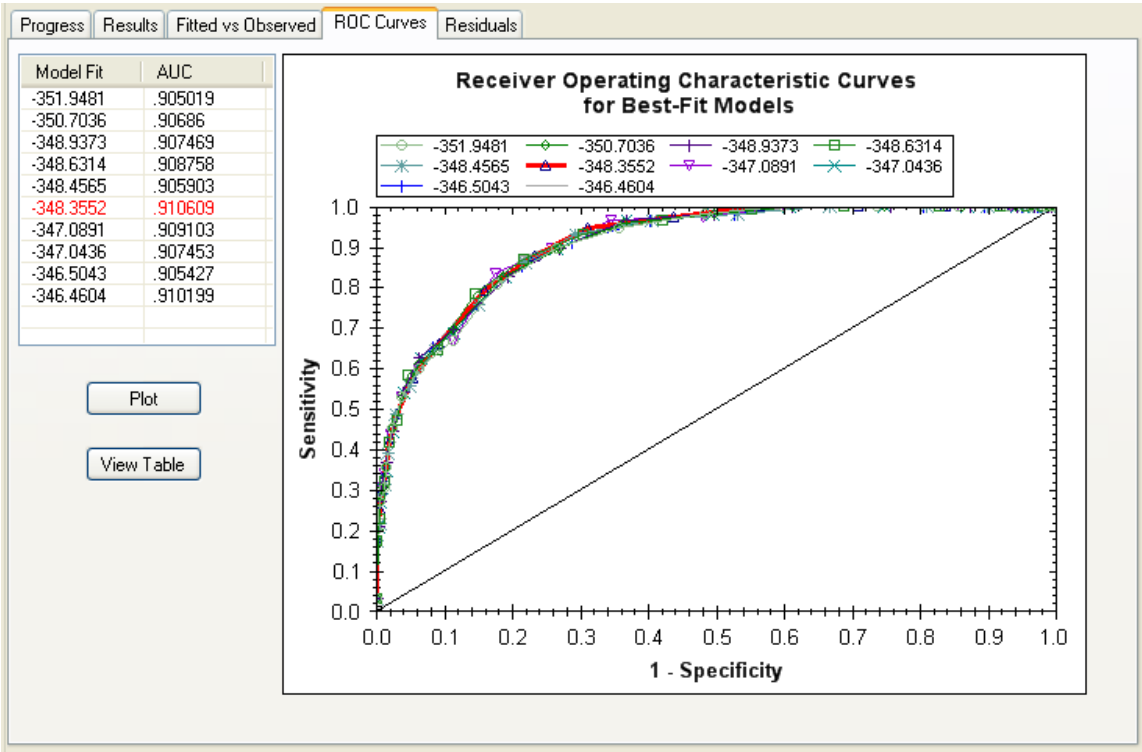


Figure 36. The ROC curves and AUC table for the Best Fit models

7.6 Viewing X-Y Scatterplots

In multiple locations within $VB_{2.4}$ (Modeling, Residual Analysis and MLR Prediction), users can access a tab that allows them to view information for comparing observations to model fitted values or predictions (Figure 35). From this space, users can view four different pieces of data:

1) A plot of fitted values/predictions versus observations: "Pred vs. Obs"
2) A table summarizing model errors (false negatives/false positives) as the decision criterion (DC) varies across the range of the response variable: "Error Table: DC as CFU"
3) A plot of the percent of probability of exceedance (calculated based on the current DC) versus observations: "% Exc vs. Obs"
4) A table summarizing model errors as the percent of probability of exceedance is varied: "Error Table: DC as % Exc"

These four are chosen with the drop-down menu at the top left corner of the form. On both of the two plots, a right-button click in the plot area shows a menu of functions for saving, copying, printing or manipulating the plot view. The plot area can be zoomed and un-zoomed: left-button mouse drags an area for zooming in; with right-button click, select "Un-Zoom" or "Set Scale to Default" to see the entire data set. To pan to an area of the plot not in view, hold the Shift key down and use the left mouse button to drag the view. To view (x,y) values of any data point, hover the cursor over the data point. If the

information does not appear, right-click on the graph and make sure “Show Point Values” is selected.

In regards to interpretation of these plots, the green (Regulatory Standard) and blue (Decision Criterion) lines permit model evaluation and provide information on which to base a DC to be used for predictive purposes. On the plots, false positives represent data points in the upper left quadrant of the graph, in which the model fits/predictions exceed the DC, but observations are below the RS. In such cases, a beach advisory would be incorrectly issued based on the model prediction, leading to potential economic losses. False negatives (points in the lower right quadrant) represent a potentially more serious scenario: model fits/predictions below the DC and observations that exceeds the RS. In other words, swimming at the beach may have been allowed when it should have been prohibited due to elevated FIB concentrations.

A model that produces no false positives or false negatives would be an ideal decision tool, but this is often unattainable with real data. Examining the two tables (#2 and #4 mentioned above) on this subtab should allow users to set a robust DC (either using units of the actual response variable or a percentage probability of exceedance) that minimizes both errors. Note that in most cases, the RS is set based on federal or state law and should not be adjusted by the user, however, the user is free to adjust the DC to minimize false negatives and false positives.

7.7 ROC Curves

In addition to time series and scatterplots which show results for an individual model, users may also compare all “Best Fits” models using the ROC Curves tab (Figure 36). A Receiver Operating Characteristic curve shows a model’s true positive rate (sensitivity) plotted against its false positive rate (1 - specificity) as a decision threshold varies between the model’s minimum and maximum predicted values. Models can then be compared using the area under their ROC curves (AUC). Models having the largest AUC values perform best over the entire decision space.

The model with the largest AUC appears in red text in the ROC tab’s model list. A single ROC may be plotted by selecting a model in the list and clicking “Plot.” Multiple models can be selected in the usual Windows fashion with Shift-Click (select all items between the first and second selection) or Control-Click (select only the clicked items). The background cell color of models *not* selected for plot display will be gray after the “Plot” button is clicked.

Clicking “View Table” will replace the ROC plot with a table showing the false positives, false negatives, sensitivity, and specificity at every evaluated value of the Decision Criterion for a single selected model. Users need only click on a model in the list to the left of this table to see its results. The ROC plot will return to view after clicking “View Plot.”

AUC calculations are performed and curves are plotted when the “ROC Curve” tab is selected. If this tab is active and new models are subsequently built, leaving this tab and then returning will generate the new plots and AUC values.

7.8 Residual Analysis

Users may click the "Residuals" subtab to view information about residuals of the selected model. There are three additional tabs on the Residuals tab: Residuals vs Fitted, Fitted vs Observed, and DFFITS/Cooks (Figure 37).

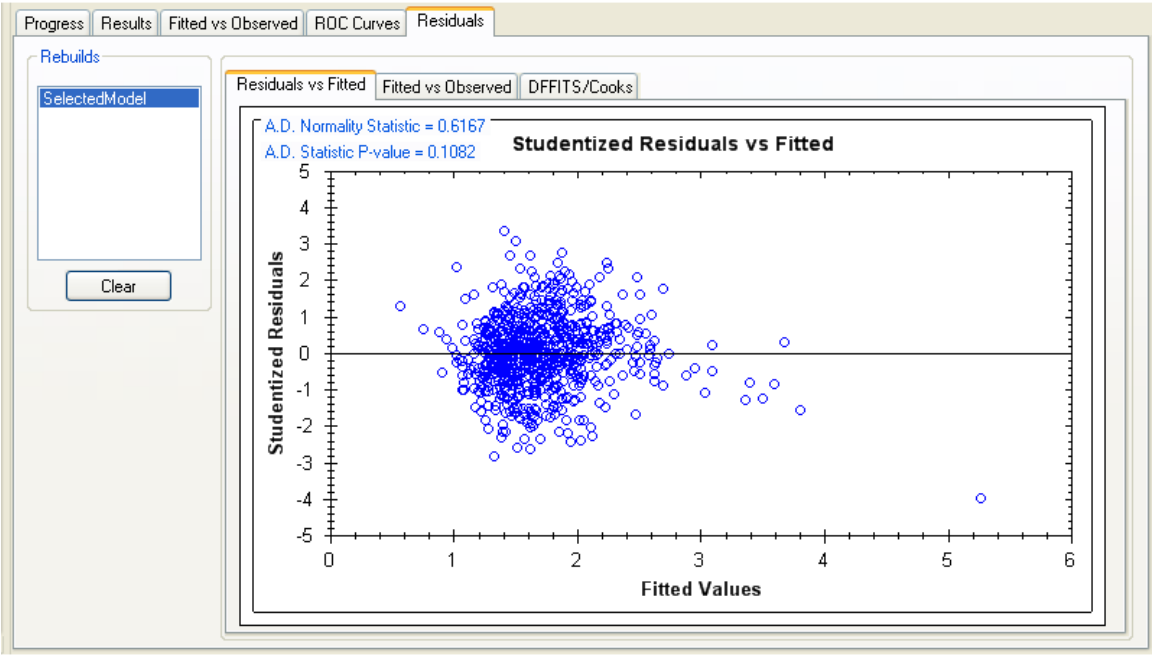


Figure 37. Information available on the Residuals subtab, including a plot of studentized residuals versus fitted model values, the Anderson-Darling residual normality test, and regression statistics

The Residuals vs Fitted tab shows a plot of the studentized residuals versus their fitted model values. In the upper-left corner of the plot (Figure 37), the Anderson-Darling Normality Statistic (http://en.wikipedia.org/wiki/Anderson-Darling) is shown with its significance (p-value). Linear regression assumes normally-distributed residuals, so if this A-D normality test fails (the p-value is less than 0.05), the user should 1) transform the response variable, 2) transform some of the IVs, or 3) consider deleting high leverage observations, which can be done from the DFFITS/Cooks tab.

On the DFFITS/Cooks tab, observations are sorted by the largest (absolute value) measure in a grid (Figure 38). At the lower left, the user can use radio buttons to toggle between DFFITS or Cook's Distance values, as well as changing their view from a grid of the sorted values to a plot of the DFFITS or Cook's Distance values versus the Record ID (Figure 39). Data points with very large DFFITS/Cook's Distances (i.e., lie outside the horizontal red boundaries on the graph) distort the fitted values and standard deviation of the regression coefficients.

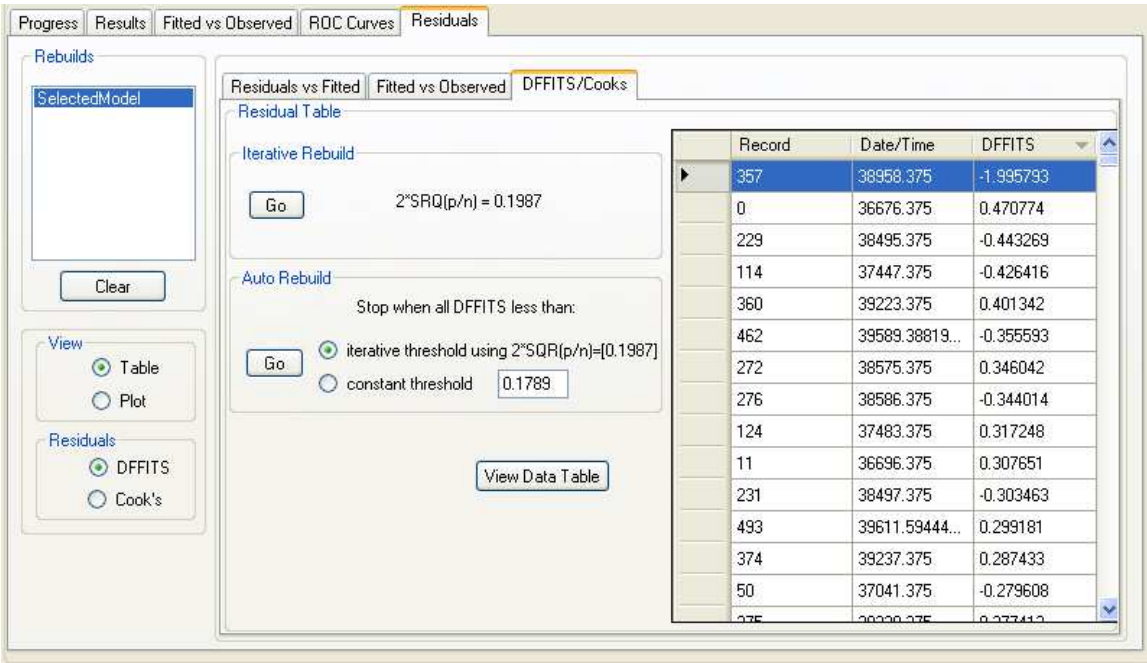


Figure 38. A table of the DFFITS scores of the residuals

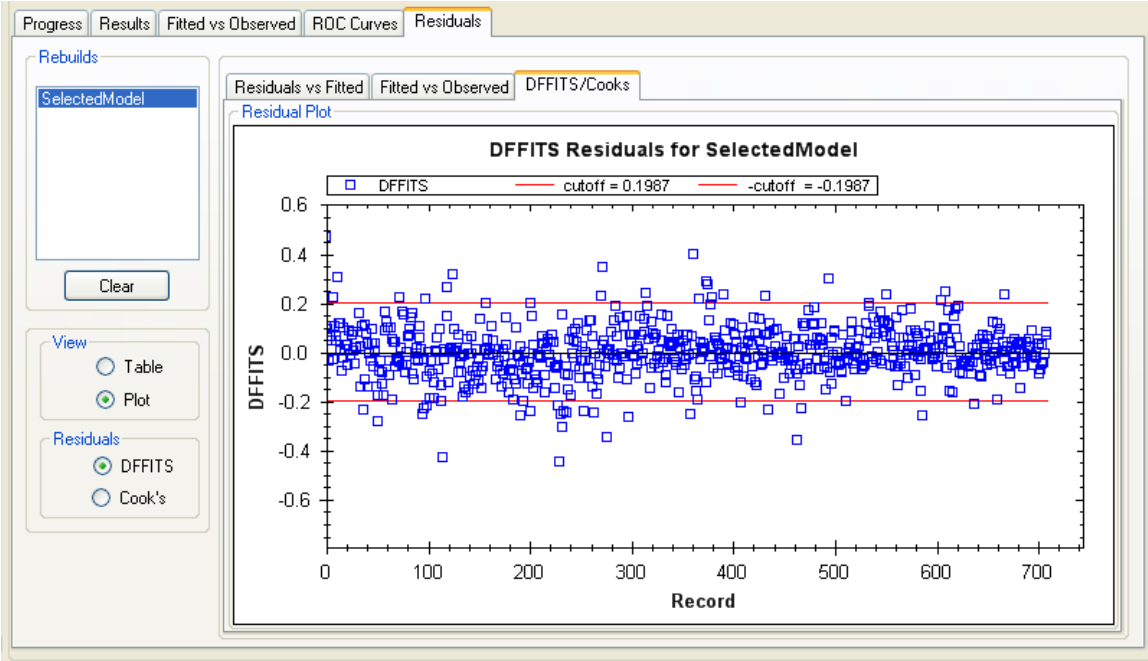


Figure 39. A plot of the DFFITS scores of the residuals

When the grid of DFFITS/Cooks values is visible, clicking the Iterative Rebuild "Go" button removes the observation with the largest absolute value DFFITS/Cooks, re-fits the regression, and calculates new DFFITS/Cooks for the remaining observations. This model is named "Rebuild1," and added to the "Models" window at the top left of the screen. Clicking on the Iterative Rebuild "Go" button again would produce a model

called “Rebuild2,” which is calculated after removing the observation with the largest absolute value DFFITS/Cooks remaining in the dataset. The user can continue to click “Go” and remove observations with the largest remaining DFFITS/Cooks, thus creating “Rebuild3,” “Rebuild4,” “Rebuild5,” etc. VB will not allow a user to delete any observations if 10 or fewer observations remain in the dataset.

Whenever a “rebuild” is created by pressing “Go,” the information displayed in the Variable and Model Statistics tables, as well as the plots and information on the Residuals subtab, is automatically updated to reflect this new model (even if another model is highlighted in the “Best Fits” window). However, the user can select any model in the “Best Fits” window to view its associated data and plots.

The user has complete freedom to carry out the outlier removal process while toggling back and forth between DFFITS and Cook’s Distance measures. For example, the first removal can be based on a DFFITS value, the next removal can be based on a Cook’s Distance, the next two removals can be based on DFFITS, etc. If the user wishes to clear the models from the “Rebuilds” window, simply click the “Clear” button.

Rather than using Iterative Rebuild, the user has two additional choices for Auto Rebuild, both of which remove all observations above some threshold. The “iterative threshold” choice bases removals on a threshold that is updated every time an observation is deleted. For DFFITS, this threshold is $2*(p/n)^{0.5}$, where p is the number of IVs in the model and n is the current number of observations in the dataset. For Cook’s Distance, the threshold is 4/n.

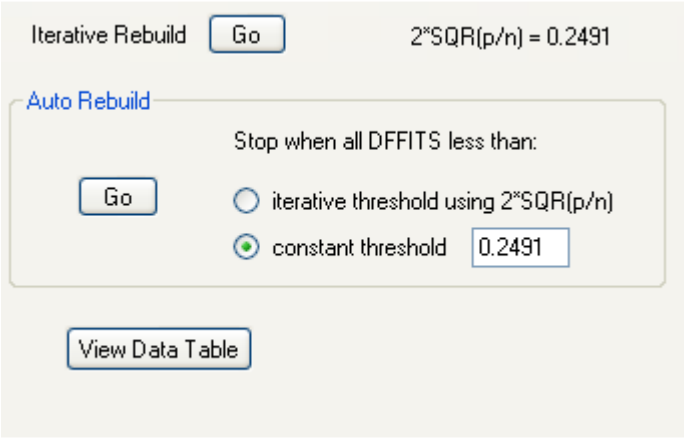


Figure 40. DFFITS/Cook’s Distance controls for removing highly influential data points

In the “iterative threshold” process of “Auto Rebuild,” step one is to check if any DFFITS/Cooks are above the threshold; if so, $VB_{2.4}$ removes the observation with the largest absolute value DFFITS/Cooks and then recalculates the regression model, the DFFITS/Cooks, and the threshold (because n has been reduced by 1). VB then checks to see if any of these new DFFITS/Cooks are above the new threshold. If so, the process repeats. VB will continue until no DFFITS/Cooks remain that exceed the current threshold, or until half of the dataset has been removed, whatever comes first. For example, if a dataset has 100 observations, $VB_{2.4}$ will allow 50 to be removed before it breaks out of the Auto Rebuild removal loop. At that point the user can click the Auto Rebuild “Go” button again to potentially remove another 25 observations of the remaining 50. We note that, in practice, one should not remove more than about 5% of the original dataset as outliers; the need to remove more indicates a poor MLR fit and

warrants a different analytical technique. Indeed, with normally distributed data, we expect 5% of the observations to be fit relatively poorly by the model.

Using the "constant threshold" Auto Rebuild option differs from the "iterative threshold" only in that the threshold remains static (i.e., the value the user types into the input box) regardless of how many observations are deleted. Updated DFFITS/Cooks are still calculated after every removal event. $VB_{2.4}$ will also stop this process if half the number of starting observations has been deleted. There is an upper limit to the number that can be entered into the "constant threshold" input box (DFFITS = 3, Cook's Distance = 16/n).

Upon completion of the Auto Rebuild process, multiple models may have been added to the "Models" window. For example, if 10 observations were removed, then "Rebuild1" through "Rebuild10" will appear in the "Models" window.

If a user has interest in both DFFITS and Cook's Distances as outlier metrics, we suggest one of the following methods:

1) *To see if the two criteria would produce different results:*
   Apply DFFITS removal to your model of choice. Note the results and then clear the Residual tab using the "Clear" button. Next perform a removal process based on Cook's Distance and compare the results.

2) *To filter out observations that offend either DFFITS or Cook's Distance criteria:*
   Run DFFITS removal on the model (i.e., remove all observations above your specified DFFITS threshold), then click the Cook's Distance subtab and perform additional outlier removal based on its threshold. After this process, remaining observations are fine from the perspective of both metrics.

Note that when the user wants to move from the Modeling tab to the MLR Prediction tab, the model that will be carried forward is the last model clicked in either the "Best Fits" window or the "Rebuilds" window. It's easy to confirm which model will be carried forward by checking the title of (and data within) the "Variable Statistics" and "Model Statistics" tabs. Also note that any observations removed from the "Residuals" tab are not removed from the primary dataset shown on the "Data Processing" tab.

Viewing the Data Table

From the DFFITS/Cooks subtab, users can click on "View Data Table" to display a history of the observation removal process for the selected model. From this window, users may export the dataset for external use or re-importation into $VB_{2.4}$.

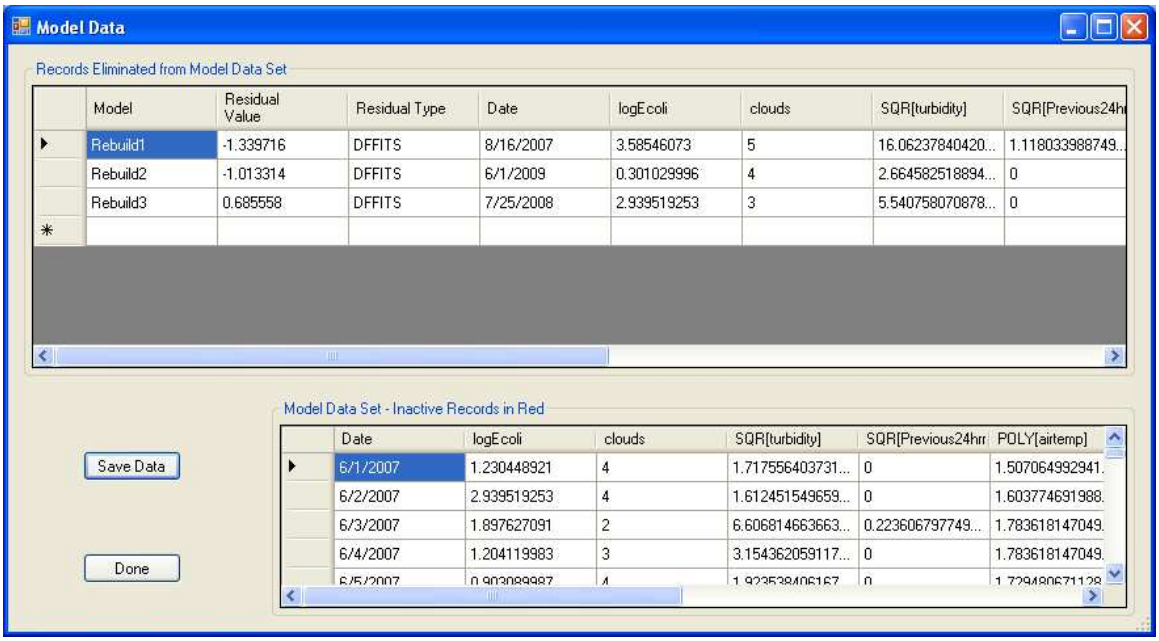


Figure 41. "View Data Table" window for examining the dataset after removal of influential data points

The "Fitted vs Observed" plot is the same as introduced in Section 7.6. There are two plots and two tables to examine, along with controls to modify the Decision Criterion (blue horizontal line) and Regulatory Standard (green vertical line), to judge effects these changes have on model outcomes (false positives, false negatives, sensitivity, specificity, etc.).

Figure 42. Fitted vs Observed plot on the Residual subtab with model evaluation threshold control and model evaluation statistics

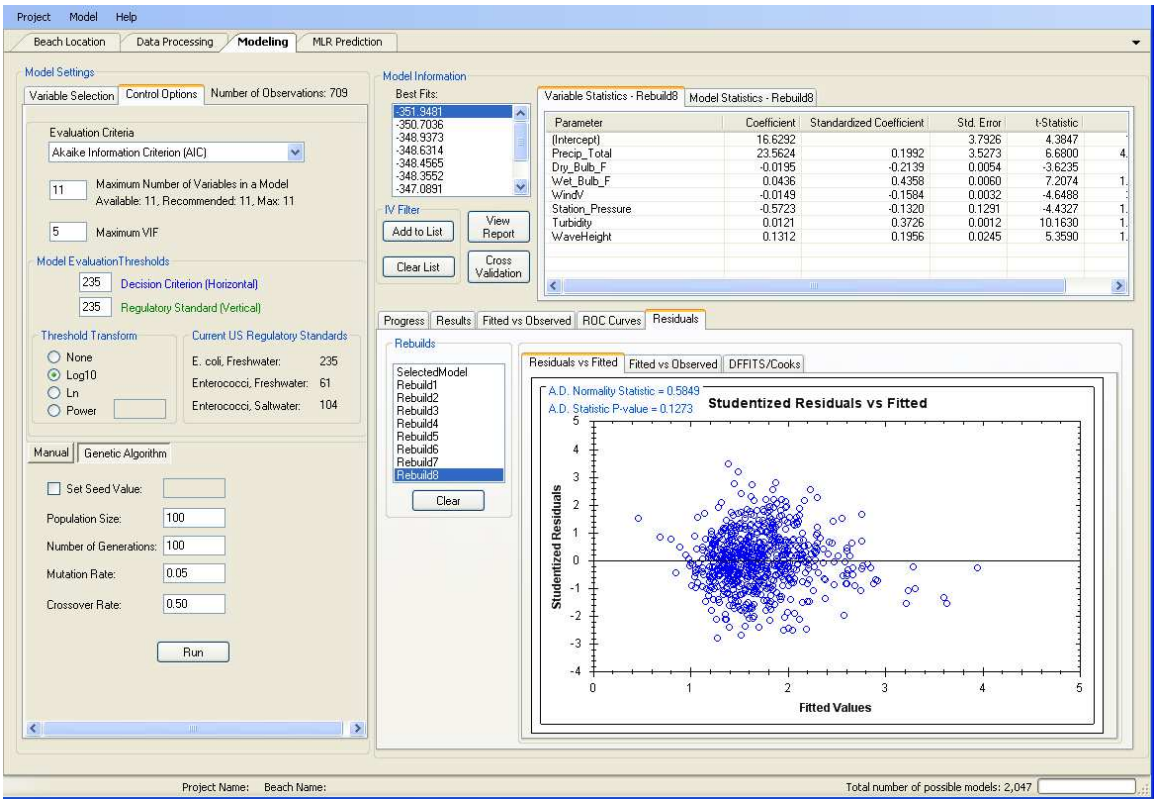


Figure 43. Residuals interface showing a list of rebuilt models resulting from observation deletions, and the associated statistics and residual plots for these rebuilds

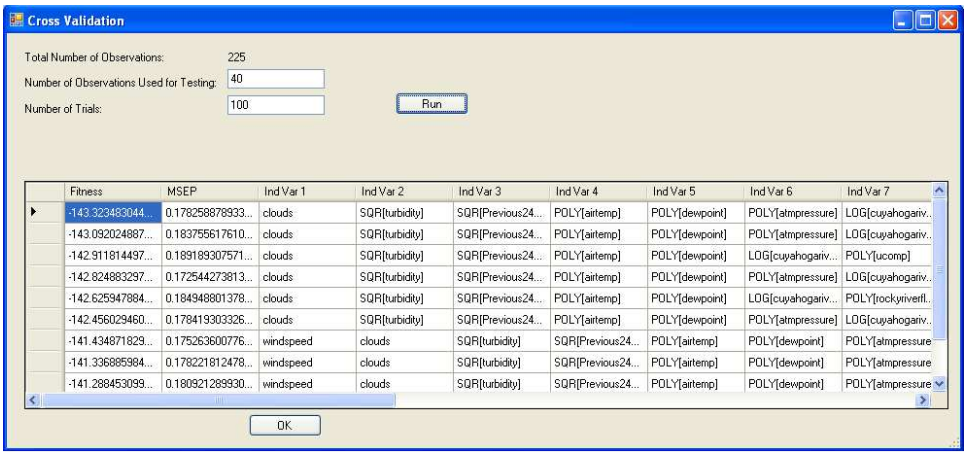
7.9 Cross-Validation

Clicking the "Cross-Validation" button on the Modeling tab brings up a sub-screen. On it users can set two parameters: sample size for the *testing* data (T) and number of random samples (R) taken. When cross-validation is started, a random sample of size T is taken from the modeling dataset and set aside. Each "Best Fits" model is then re-fit to the remaining *training* data. The IVs in each model stays the same, but the regression coefficients are adjusted to reflect the least-squares fit to the *training* data. The Mean Squared Error of Prediction (MSEP) is then calculated based on the T *testing* data points for each candidate model. The process (taking a random *testing* sample; re-fitting regression coefficients for the ten candidate models based on the *training* data; using the re-fit models to make predictions; and computing 10 MSEP values) will be done R times. A table will show average MSEP values for each candidate model.

Cross-validation is a widespread, useful technique for examining the predictive power of models, i.e., their ability to make predictions for data they have not seen before. For users wishing to emphasize the predictive ability of a potential model, cross-validation allows them to evaluate which candidate model consistently makes the best predictions (i.e., has the lowest MSEP). Note that the PRESS statistic $VB_{2.4}$ provides as a model evaluation criterion is a cross-validation statistic with T set to 1. The PRESS algorithm removes one observation at a time from the dataset, re-fits the model regression coefficients, and then calculates the squared residual for the removed observation. It

does this once for every observation in the dataset to compute the model's PRESS value -- a confined look at a model's predictive potential.

Recommended values to enter for the observations used for testing are approximately 25% of the total number of observations and 500-1000 trials.



Fitness	MSEP	Ind Var 1	Ind Var 2	Ind Var 3	Ind Var 4	Ind Var 5	Ind Var 6	Ind Var 7
-143.323483044...	0.178258878933...	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure]	LOG[cuyahogariv..
-143.092024887...	0.183755617610...	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure]	LOG[cuyahogariv..
-142.911814497...	0.189189307571...	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	LOG[cuyahogariv...	POLY[ucomp]
-142.824883297...	0.172544273813...	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure]	LOG[cuyahogariv..
-142.625947884...	0.184948801378...	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	LOG[cuyahogariv...	POLY[rockyriverfl..
-142.456029460...	0.178419303326...	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure]	LOG[cuyahogariv..
-141.434871829...	0.175263600776...	windspeed	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure
-141.336885984...	0.178221812478...	windspeed	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure
-141.288453099...	0.180921289930...	windspeed	clouds	SQR[turbidity]	SQR[Previous24...	POLY[airtemp]	POLY[dewpoint]	POLY[atmpressure

Figure 44. The cross-validation results for each of the 10 best-fit models

7.10 Report Generation

A text report of modeling results can be generated, copied to the system clipboard, or saved to a text file using the "View Report" button. Users can view the report within $VB_{2.4}$ by selecting the desired models and clicking on "Generate Report for Selected Models." The report contains descriptive statistics for each model variable and model evaluation statistic. Any number of best-fit models can be selected for reporting.

A recommended approach to saving the information in an external application is to copy the report to the clipboard (with the "CopytoClipboard" button) and paste it into a rich-text application like MS Word, Write or WordPad. NotePad or other text editors will work, but column formats will likely be lost and make the report difficult to interpret.

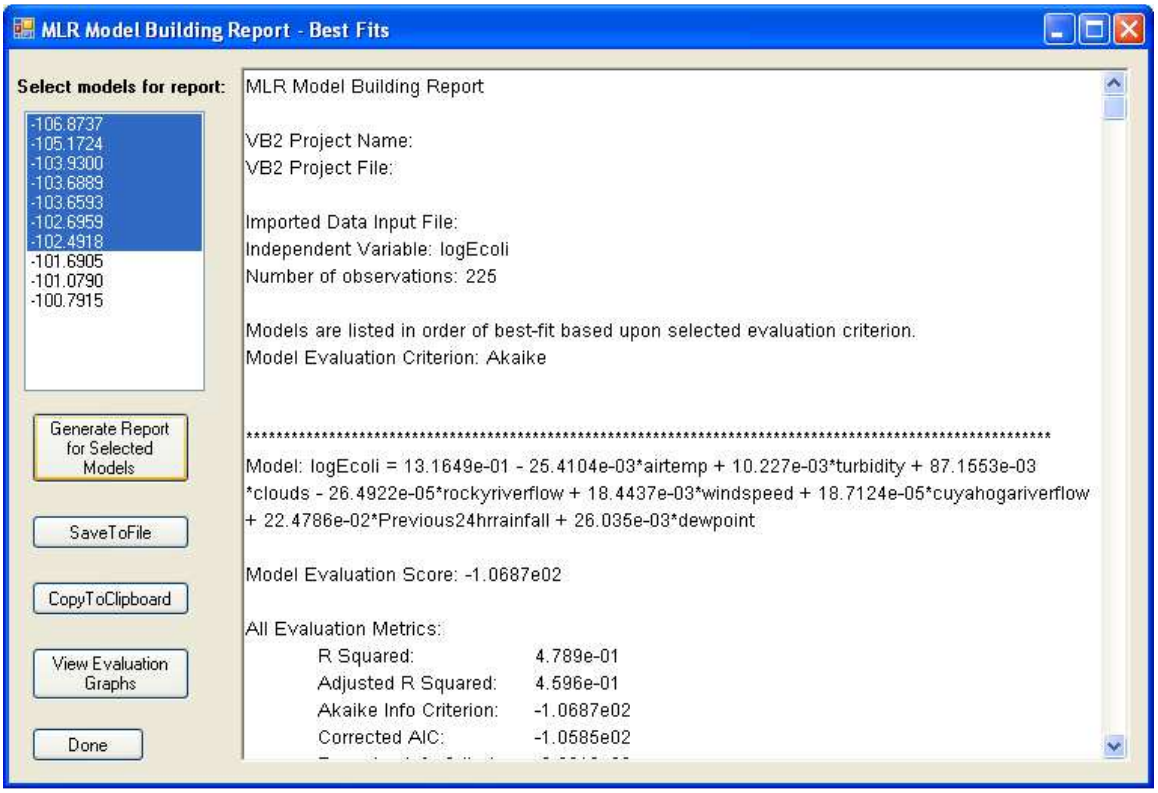


Figure 45. A text report generated on the modeling results

Comparative bar graphs can be displayed to view evaluation criteria for all top models. Click on "View Evaluation Graphs" to see these plots. Hover the mouse over any plot to display the relevant evaluation criteria and hovering over any bar displays the associated model. Note that the evaluation criteria graphs are scaled to emphasize differences between the model scores although the difference may, in fact, be quite small. With the cursor over any graph, right-mouse click and select "Set Scale to Default" to view the un-scaled graph.

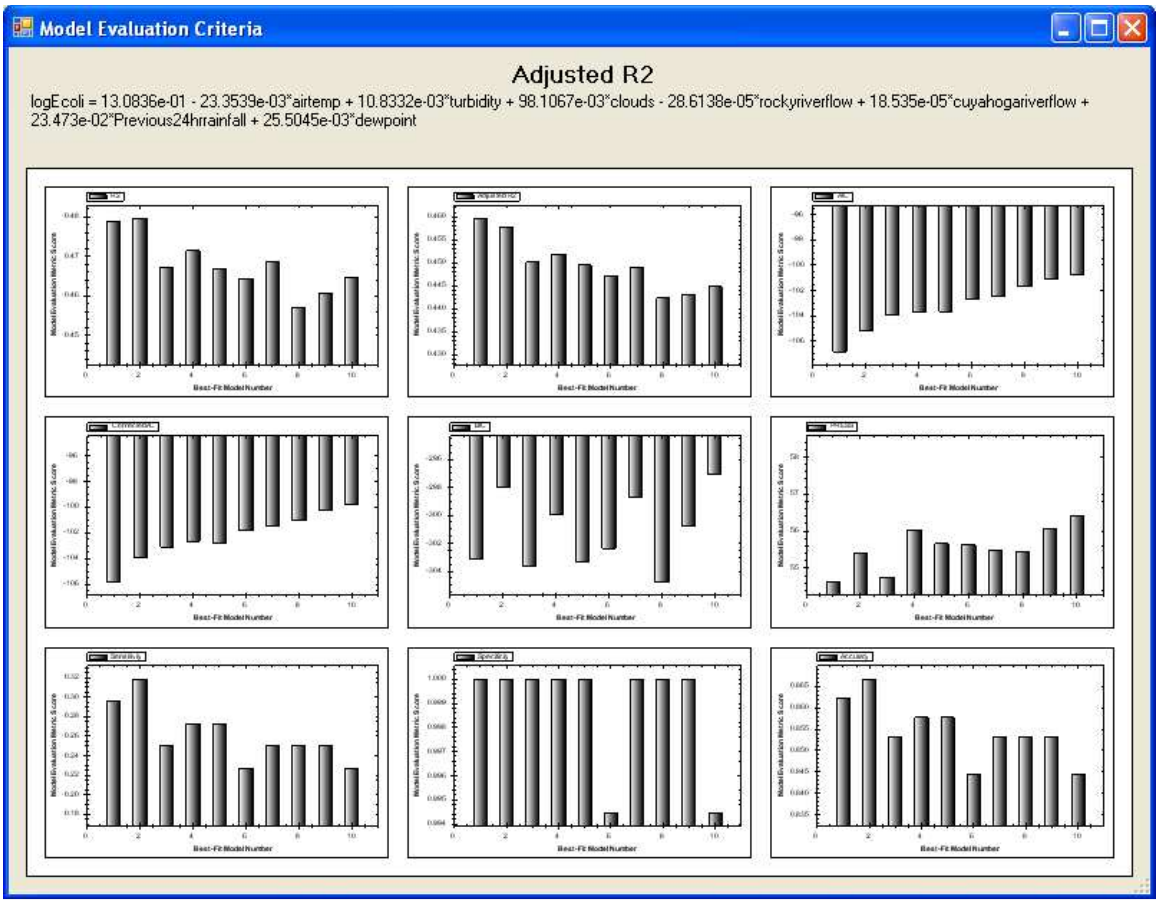


Figure 46. Plots of the various model evaluation metrics for the 10 best-fit models

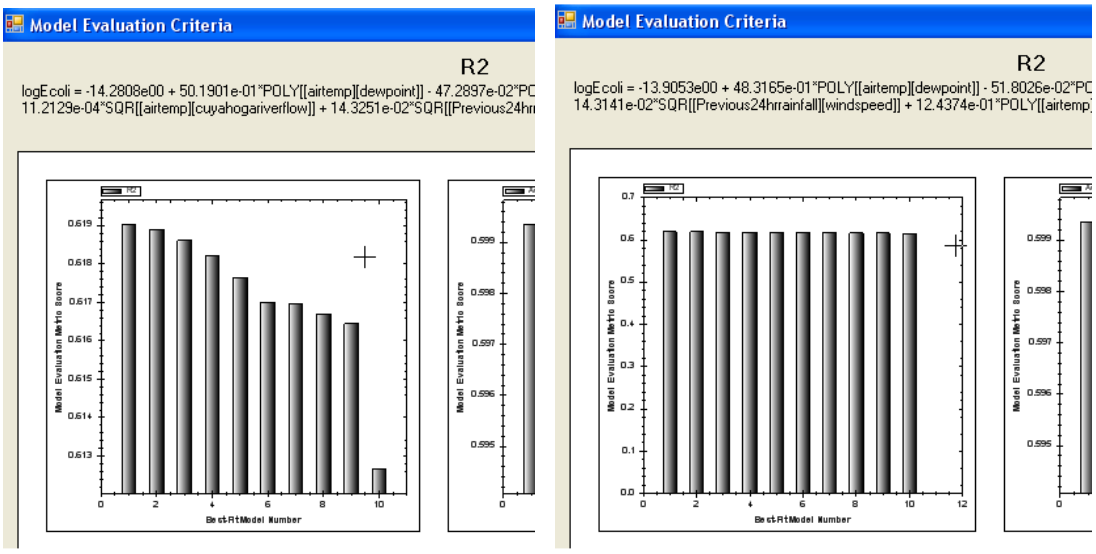


Figure 47. Scaled versus un-scaled views of selected model evaluation criterion

8. PREDICTION

The Prediction interface allows users to estimate or predict FIB concentrations with a selected regression model. Whether a user was previously on the Modeling tab (with a model selected in "Best Fits") or on the Residuals tab (with a model selected in "Models"), the interface of the MLR Prediction tab will look the same.

8.1 Model Statement

At the top is the linear expression for the chosen model, with values of the regression coefficients and names of each IV in the model (Figure 48).

8.2 Model Evaluation Thresholds

There are input boxes for the Decision Criterion (DC), Probability of Exceedance, and Regulatory Standard (RS). Setting these allows model predictions to be evaluated and model specificity, sensitivity, and accuracy to be calculated. When users first arrive at the Prediction tab, values of the DC and RS will be set to what was on the Modeling tab. The "Threshold Transform" button tells $VB_{2.4}$ how to transform the DC and RS for comparison to model predictions and observations. If a transformation definition was set for the response variable during data processing (either manually by the user or automatically by transforming the response), that definition will be set here as the default. Users should be aware that changing the threshold transform definition can cause problems when comparing modeling predictions to observations. Caution should be exercised.

There are also two red labels indicating the response variable (RV) transform and the observation transform. These inform the user of the currently specified transformation types for these two data pieces. The RV transform was set back on the data processing tab, while the observation transform is set on this tab by right clicking on the column header of observations (in the middle section of the lower table). If the user imports or types in untransformed observations, they do not have to do anything - $VB_{2.4}$ assumes by default that observations are untransformed. If the user imports/enters log10-transformed observations, for example, then they need to right-click on the observations' column header and choose "Log10" from the list of choices. This gives $VB_{2.4}$ the information it needs to correctly compare observations to model predictions.

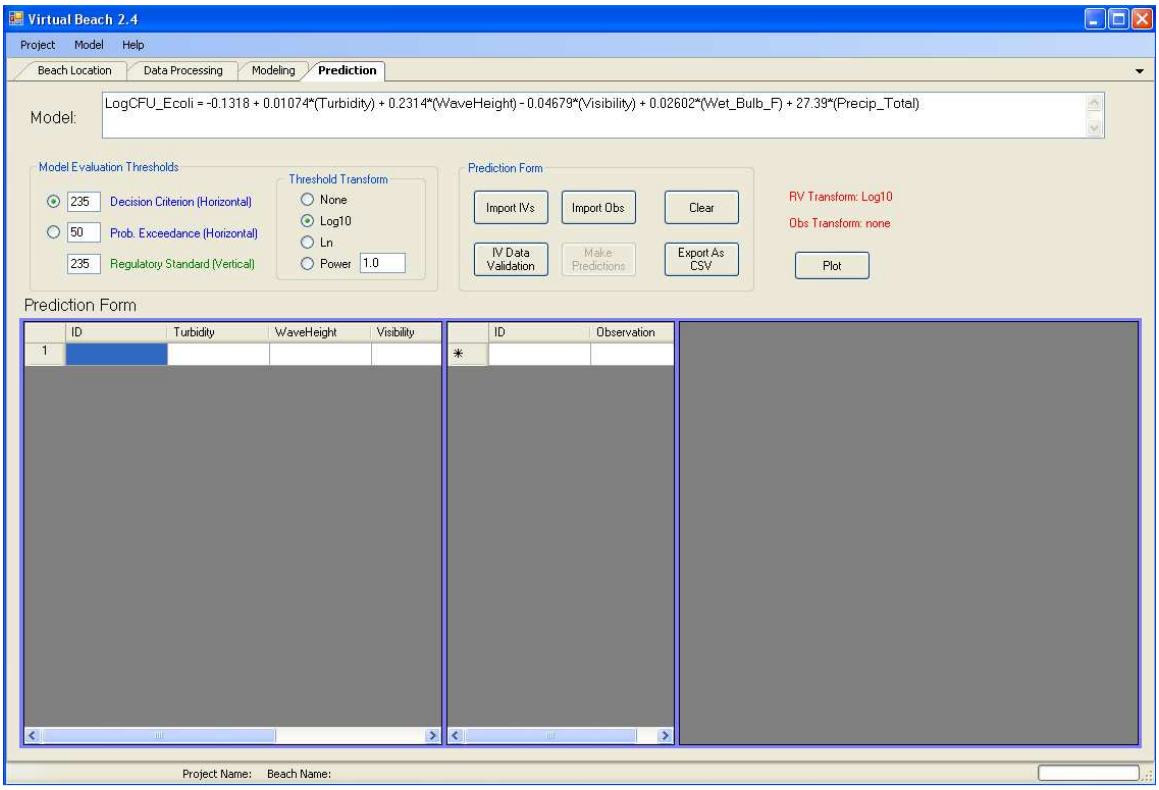


Figure 48. The Prediction interface

8.3 Prediction Form

Most of the prediction form is in three separate data panels: the left panel holds IV data; the middle panel is for observational data, e.g., lab results of FIB concentrations; and the right section shows model predictions and evaluation metrics. Each panel also contains a column for a unique ID for each row of data (e.g., the date that data were collected). The panels have separate horizontal and vertical scroll bars that become visible if the number of rows or columns exceeds the viewable area. The three panels independently scroll horizontally, but scroll as a group vertically. Panels can be re-sized by clicking and dragging the blue vertical partitions. Order of the columns in the left and right panels can be changed by clicking and dragging the column headers left or right.

Users can import IV and observational data from a file using “Import IVs” and “Import Obs” buttons in the “Prediction Form” button bank located in the middle right of the screen, or users can type data into the input grids. Either way, they should be certain that the entered IV data are in the same units as those used to build the model.

Depending on which model was selected for prediction, the IV panel will have one column for every unique IV that appears in the model, plus a column for the row’s unique ID. When a data file is imported with the “Import IVs” button, a “Column Mapper” window opens. This window allows users to tell $VB_{2.4}$ which columns in the imported datasheet should be used for the row IDs and each IV found in the model. By default, the first column of the imported file maps to the ID field, but users can choose another column if needed. If a column in the imported spreadsheet has an identical name to an IV in the model, that column will be automatically selected by $VB_{2.4}$ as the appropriate one for that IV.

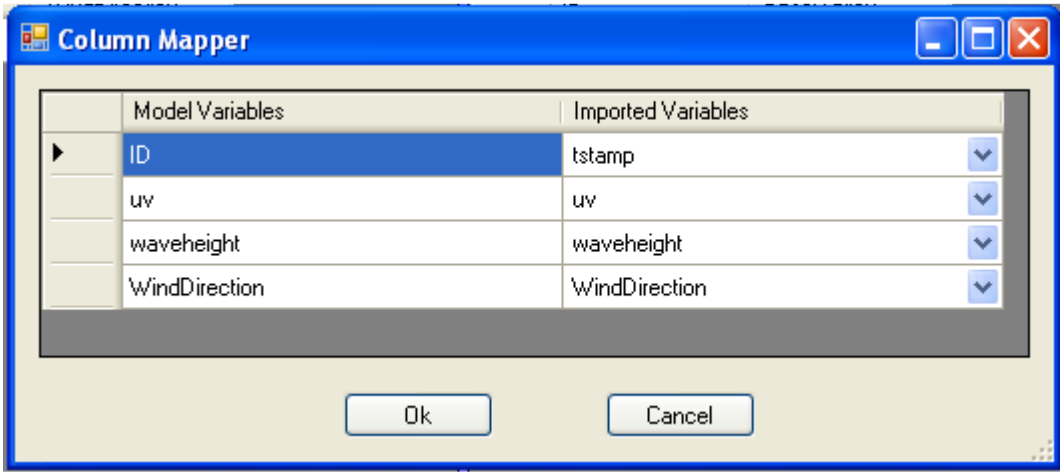


Figure 49. Importation of IV data using the "Column Mapper" window

As with IV data, observational data can be typed into the middle panel or imported using "Import Obs." For observational data, only two columns are needed: row IDs for every observation and the actual observations. A "Column Mapper" window appears when observational data are imported from a file. After they have been imported or manually entered, users can specify the observation type for a proper comparison to model predictions. This is done by right-clicking on the "Observation" column header and defining the transformation: none, $\log_{10}$, $\log_e$, or a power transformation. "None" is the default choice. For example, if Log10 observations are imported, the user would need to change the right-click menu choice to "Log10."

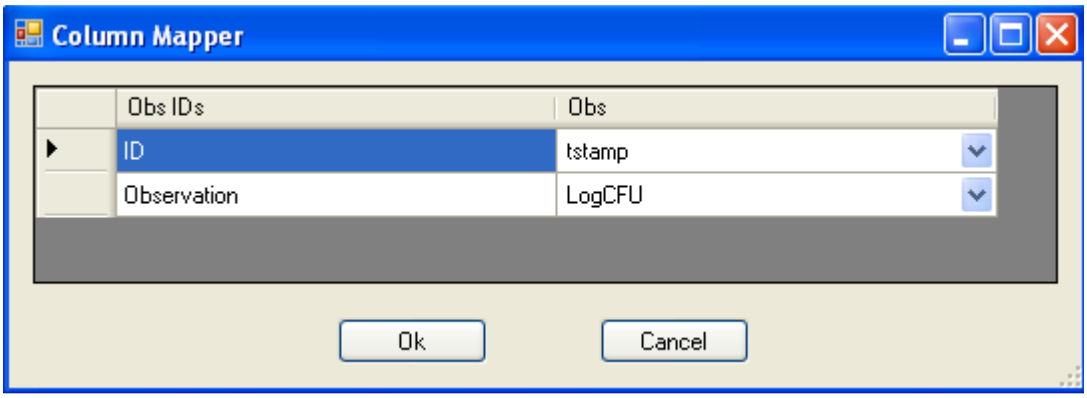


Figure 50. Importation of observational data using the "Column Mapper" window

The "Make Predictions" button remains disabled until the IV data (imported from a file or manually typed) are validated using the "IV Data Validation" button. This scan ensures there are no blank cells or non-numeric data in the IV columns of the IV data panel and checks that every row ID is unique (non-numeric data are allowed for the ID column). This validation scan window is very similar to the validation scan window sin the Data Processing tab; however, "Delete Column" is not a choice. "Replace With" and "Delete Row" are the only ways to deal with problems in the IV data grid.

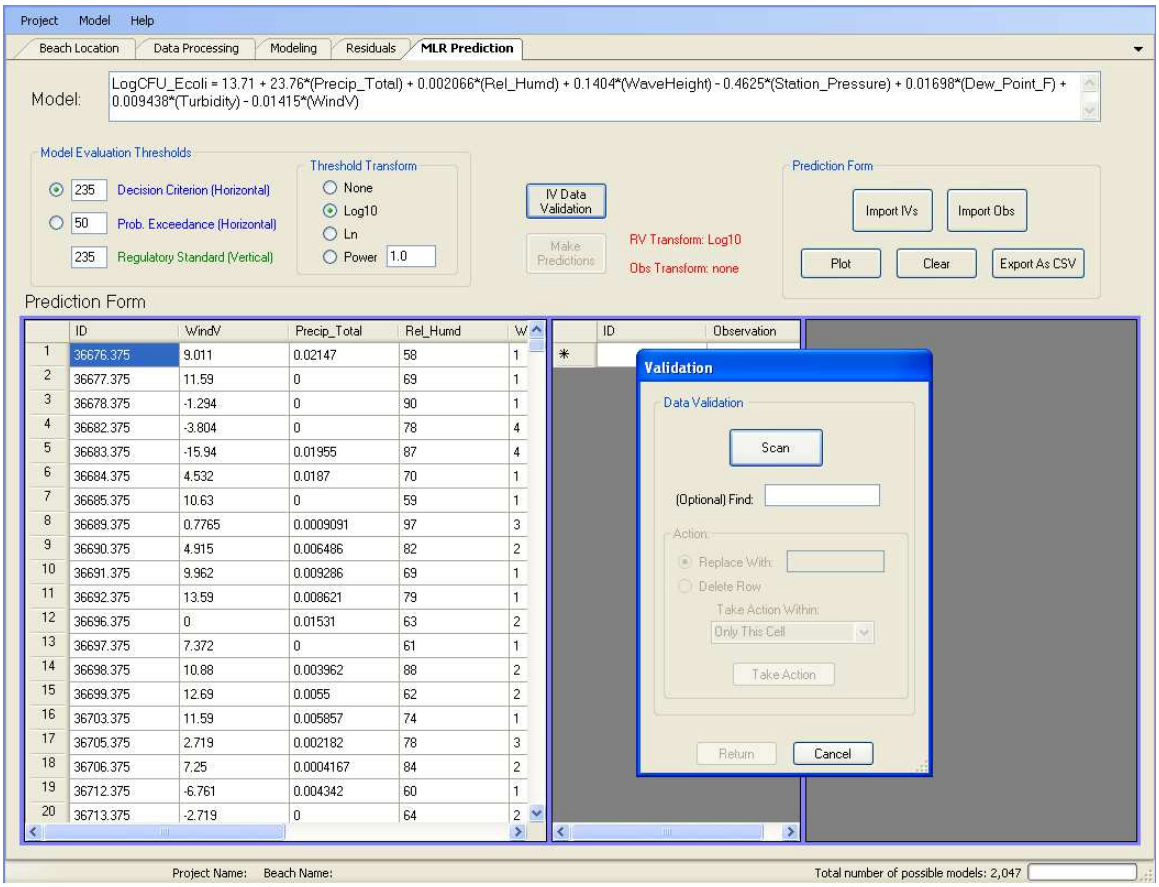


Figure 51. The IV validation window on the MLR Prediction tab

Once IV data have been validated, clicking the "Make Predictions" button will generate model predictions. Observational data need not be present to make predictions, but observations are needed for model evaluation (sensitivity, specificity, false negatives, false positives, etc.). After clicking "Make Predictions", $VB_{2.4}$ uses the model, IV data, and observational data to fill the right panel with the following data columns: ID, Model Prediction, Decision Criterion, Regulatory Standard, Exceedance Probability, and Error Type.

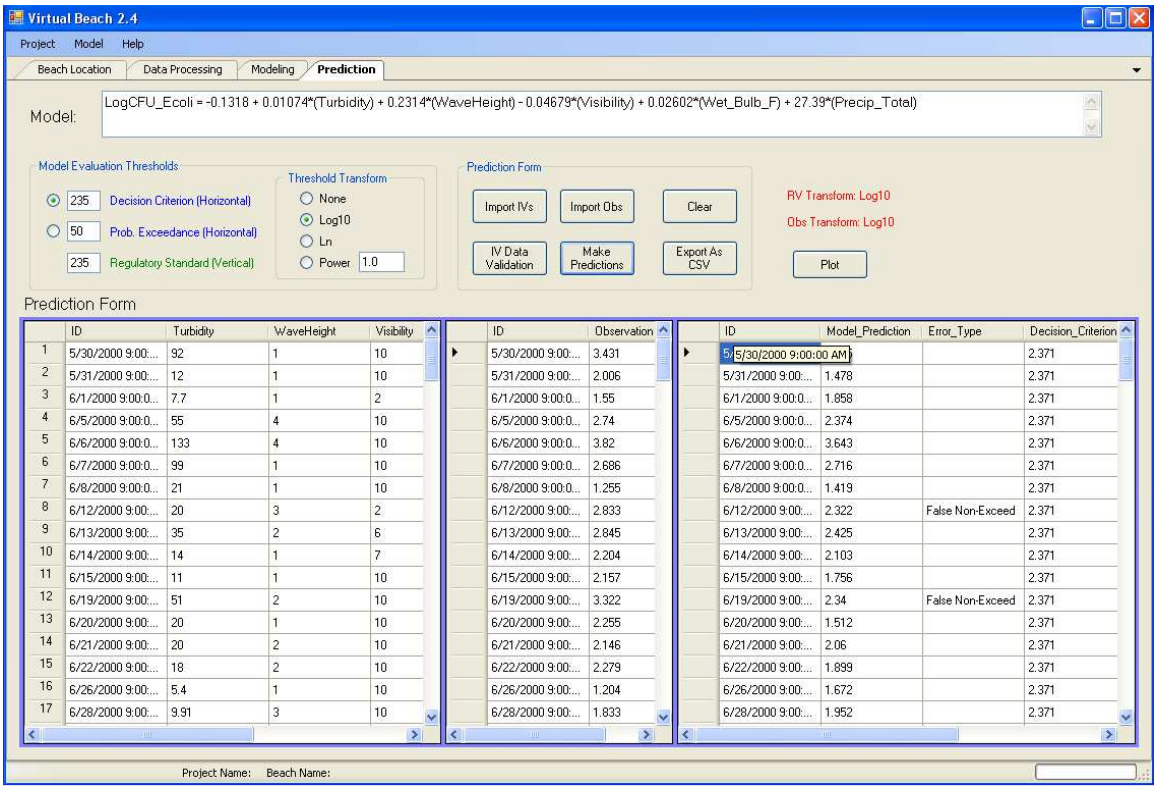


Figure 52. A prediction grid after IVs and observational data have been imported, and model predictions have been made

The ID column of the model output panel is taken directly from the ID column of the IV panel, not the observation panel. The "Make Predictions" button makes one model prediction per row in the IV data panel, regardless of how many observations are entered in the observation panel.

The Model Prediction column contains predicted values of the response variable. Right-clicking on this column header allows the user to change how the predictions are displayed in the table (as linear, log, or power units). The Decision Criterion and Regulatory Standard are values set by the user (shown in the left panel as transformed by the choice of "Threshold Transform"). The Exceedance Probability (actually the probability x 100) is defined as the probability that the model prediction will be larger than the Decision Criterion, based on uncertainty bounds (confidence intervals) around the model predictions.

To compare model predictions to observations, $VB_{2.4}$ looks at the prediction ID and attempts to find an observation in the observation panel with that same ID. $VB_{2.4}$ does not require unique IDs for each row in the observation panel, but note that a model prediction is compared to the first observation found with the same ID. When comparing model predictions to observations, an error (false exceedance or false non-exceedance) appears in the "Error Type" column.

It is important to note that accurately assessing model output depends on synchronized transformation information regarding the Decision Criterion, Regulatory Standard, model predictions, and observations. Users must be careful to ensure each value is in a comparable unit.

8.4 Viewing Plots

After predictions have been made, a scatterplot of observations versus predictions can be viewed by clicking "Plot" in the "Prediction Grid" button bank. If no observational data were entered, a message asking for observational data appears. The features and functionality of the form that appears when the "Plot" button is clicked are described in Section 7.6. The data are based on comparing model predictions (right pane of the Prediction Form) with observations (middle pane) that share the same, unique ID.

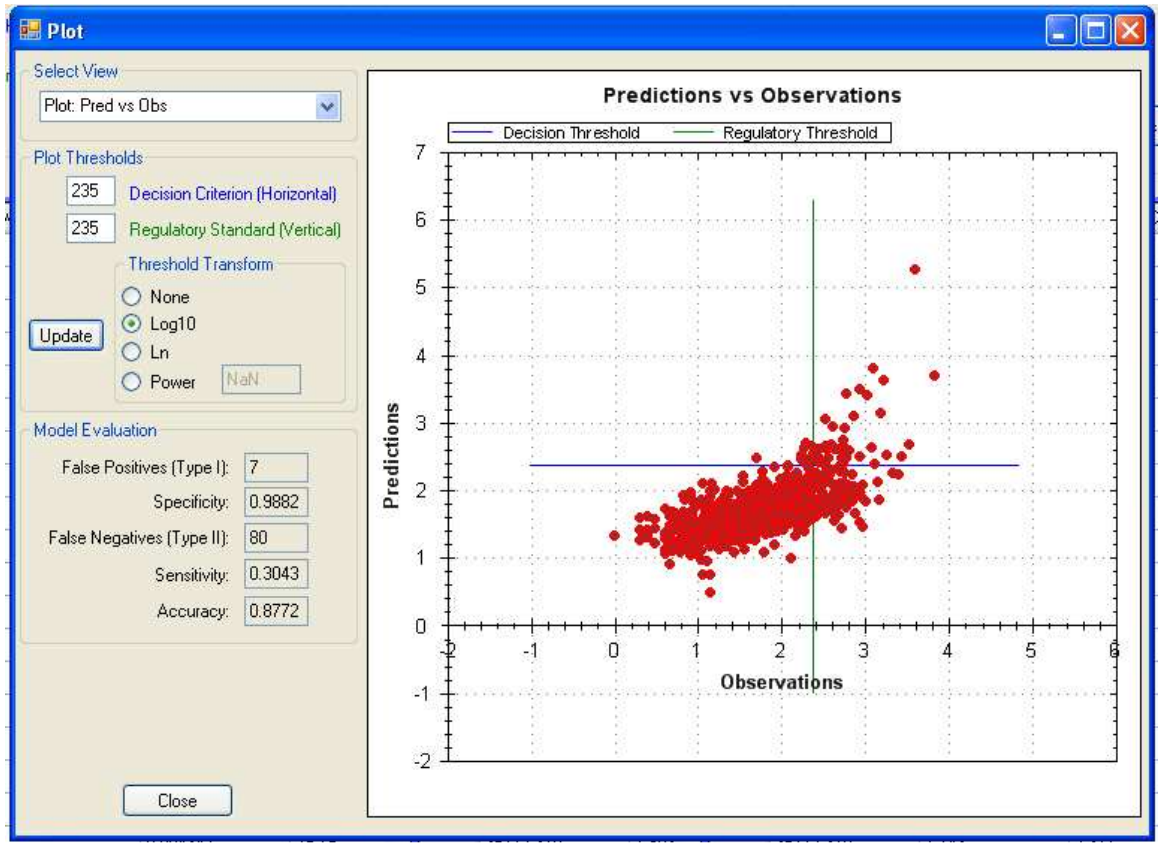


Figure 53. Prediction interface plotting of the observations versus predictions, with model evaluation threshold controls

8.5 Prediction Form Manipulation

Two other buttons are found in the "Prediction Grid" button bank. If a user wants to view the table in a spreadsheet or word processing program, "Export as CSV" saves the contents of the entire table (three panels) in .csv format. "Clear" deletes all information in the predictive table. As with most of the tabular information in $VB_{2.4}$, data in individual panels can be selected with a left click and drag. Control-c and Control-v can then be used to copy and paste the data into another application such as WordPad or Excel.

9. LATEST RELEASE

VB has undergone continuous improvement and functional expansion. In $VB_{3.0}$, released in September 2013, project management enhancements allow modeling results to be saved. The prediction interface provides access to the USGS automated data delivery system called EnDDaT for the retrieval of site-specific data such as water quality, water flow gauge readings and weather measurements. Model- building functionality goes beyond MLR to include Gradient Boosting Machines (based on decision trees), and Partial Least Squares regression.

10. USER FEEDBACK

Opinions and experiences from the user community are welcomed by the Virtual Beach design/development team. Users are encouraged to report problems, issues and likes/dislikes to:

Mike Cyterski – 706 355-8142 (cyterski.mike@epa.gov)

11. ACKNOWLEDGMENTS

We would like to thank the following people, and their institutions of employ, who generously donated their time and expertise for software testing and review of this document:

Adam Mednick
Wisconsin Department of Natural Resources
Madison, WI

David Rockwell
Cooperative Institute for Limnology and Ecosystems Research, University of Michigan
Center of Excellence for Great Lakes and Human Health
NOAA Great Lakes Environmental Research Laboratory
Ann Arbor, MI

Donna Francy
USGS Ohio Water Science Center
Columbus, OH

Wesley Brooks, Mike Fienen, and Steve Corsi
USGS Wisconsin Water Science Center
Madison, WI

Fran Rauschenberg and Richard Zepp
Ecosystems Research Division
National Exposure Research Laboratory
USEPA
Athens, GA