Bridge: a GUI Software for Genetic Risk Prediction Research

Chengyin Ye
cye@epi.msu.edu

April 11, 2013

Abstract:

The `Bridge` is a graphical user interface(GUI) package developed under R environment. This package is built for both designing and analyzing a risk prediction model. In the design stage, it provides an estimated classification accuracy of the model using essential genetic and environmental information gained from public resources and/or previous studies, and determines the sample size required to verify this accuracy. In the analysis stage, it adopts a robust and powerful algorithm for forming the risk prediction model. The package was developed based on the optimality theory of the likelihood ratio and therefore theoretically could form a test with the highest performance. It can be used to handle a relatively large number of genetic and clinical predictors, with consideration of their possible interactions, and so is particularly useful for forming a risk prediction model under a common complex disease scenario.

1 Background

The translation of human genome discoveries into health practice represents one of the major challenges in the coming decades. The use of emerging genetic knowledge for early disease prediction, prevention and pharmacogenetics will advance future genomic medicine and lead to more effective prevention and treatment strategies. Among those, disease prediction based on genetic and clinical information is the first step in translating genomics into health. It assesses an individual's risk of future disease, so that early preventive interventions can be adopted to reduce morbidity and mortality. For this reason, studies to assess the combined role of genetic and

clinical information in early disease prediction represent a high priority, as manifested in the multiple risk prediction studies now underway.

The yield from these studies can be enhanced by adopting powerful and computationally efficient study design and analytic tools. We had previously developed an optimal ROC curve (O-ROC) method, to quickly evaluate new genetic and clinical findings for potential clinical practice by designing a new risk prediction model, estimating its classification accuracy, and calculating the sample size needed for further investigation of the model [1].

If, in the design stage, a proposed risk prediction model appears to be superior to existing models, or if it reaches a desired accuracy level, it may worth developing further for clinical use. To further study the risk prediction model on collected samples, we developed a forward ROC curve (F-ROC) method [2]. F-ROC builds on the optimality theory of the likelihood ratio, and is thus powerful for risk prediction analysis. It adopts a stepwise selection algorithm to efficiently deal with a large number of predictors and their possible high-order interactions, making high-dimensional risk prediction modeling feasible.

To facilitate designing and analyzing risk prediction models, we have implemented the above two methods into the graphical user interface (GUI) software, Bridge, in the statistical software environment, R. Bridge is comprised of two modules, **Test Design** and **Test Build**. The O-ROC approach has been implemented in the **Test Design** module, for the design and evaluation of a risk prediction model. The **Test Design** module uses the essential information (e.g., allele frequencies) of risk predictors from previously published studies or publically available resources to design a risk predictive model, calculating its estimated accuracy and the sample size needed to further investigate the model. F-ROC has been incorporated into the **Test Build** module. The **Test Build** module is developed for risk prediction modeling on known risk predictors, as well as for high-dimensional risk prediction based on a large number of potential risk predictors. Bridge is freely accessible online at `https://www.msu.edu/~qlu/software.html`.

2 Installing `Bridge`

The `Bridge` GUI package is developed based on `gWidgets` package. Before installing the `Bridge` GUI package, the GTK libraries and the following R packages should

[1] Lu Q, Elston RC: Using the optimal receiver operating characteristic curve to design a predictive genetic test, exemplified with type 2 diabetes. .Am J Hum Genet. 2008 Mar;82(3):641-51

[2] Ye C, Cui Y, Wei C, Elston RC, Zhu J, Lu Q: A non-parametric method for building predictive genetic tests on high-dimensional data. Hum Hered 2011;71:161-170.

be installed: `RGtk2`, `gWidgets`, `gWidgetsRGtk2` and `cairoDevice`. Note that we build the `Bridge` GUI package on R version of 3.0.0. In order to appropriate install and use `Bridge`, we suggest the user to install any R version >= 3.0.0. If you are on windows, a script is provided on our website to automatically install the GTK libraries and these R packages, as well as the `Bridge` package itself (`https://www.msu.edu/~qlu/software.html/`). You can use the following command to run the script,

```
> source("https://www.msu.edu/~qlu/software.html/install_Bridge.R")
```

3 Starting

Users can load the `Bridge` by using the following command,

```
> library(Bridge)
```

If users want to restart `Bridge` after closing the window, you need to type the following command in R console window,

```
> Bridge()
```

4 Input Data

4.1 Input Data in Test Design

The Data entered into **Test Design** module includes disease prevalence and the general information about genetic and environmental risk factors (e.g., allele frequencies), which can be obtained from many public resources and previous studies. For genetic risk predictors, four possible types of data could be used in `Bridge`: allele frequencies in cases and controls, allele's relative risk and frequencies in the population, genotype frequencies in cases and controls, or genotype's relative risks and frequencies in the population. For environmental risk predictors, two possible types of data can be used, frequencies of environmental risk predictors in cases and controls, or the combination of environmental relative risks and the distribution of environmental risk predictors in the population. By clicking the **Data Input...** option of the **Test Design** menu, users can enter the type of data he/she prefers.

4.1.1 Input Disease Prevalence

The Disease Prevalence parameter should be entered prior to all other parameters from the **Input Disease Prevalence...** option.

4.1.2 Import data from a txt file

Users also have an option to enter the data via a txt data file. The file need to be prepared in advance and can be loaded to the system through the **Import DataSet...** option. In the txt file, the field separator must be **Tab**. If both genetic variants and environmental risk predictors exist in the txt file, the genetic risk predictors should be listed in front.

The data file uses three rows for each genetic risk predictor. The first row is used for labelling different alleles or genotypes, which should use the words of either **"Allele"** or **"Genotype"**. The next two rows is used for entering the data. For data in the forms of frequencies in cases and controls, the column names should be **"Case"** and **"Control"**. For data in the forms of relative risk and population frequencies, the column names must be **"RelativeRisk"** and **"Population"**.

Similar as the genetic risk predictors, the data file also uses three rows for each enviromental risk predictor. However, the data file allows enviromental risk predictors with more than 3 categories. Users can use this option for genetic risk predictors with more than 2 categories (e.g., haplotypes).

The Figure 1 gives an example of the data file. An example file, DesignData.txt, is also freely accessible online at `https://www.msu.edu/~qlu/software.html`.

4.1.3 Input data manually

Alternatively, we can input data manually from the **Input Data Manually...** option, which could be much easy and straightforward.

4.2 Input Data in Test Build

In the **Test Build** section, users are allowed to import two datasets, one for model building and the other for model evaluation. The first dataset is required and the second one is an option. The first dataset (i.e.,training dataset) can be imported by clicking the **Import a dataset...** option under the section of **Data Input...** from the **Test Build** menu, while the validation dataset can be import from the **Import a second (validation) dataset...** option. Both datasets used should have the same format. For each individual, both disease status and genotypic information should be provided. For the disease status, the package **only** accept the dichotomous outcome,and should be coded as 1 or 0 for case and control, respectively. For the genetic markers, the current version of **Test Build** is limited to SNP data, which should be coded as 0, 1 and 2. The **Test Build** can handle missing data, and the missing genotype should be coded as -9. Only datasets importing from .txt files is

Allele	SNP1 A	SNP1a	SNP2 A	SNP2a					
Case	0.3	0.7	0.6	0.4					
Control	0.7	0.3	0.4	0.6					
Allele	SNP3 A	SNP3a							
RelativeRisk	1.5	1							
Population	0.5	0.5							
Genotype	SNP4 AA	SNP4 Aa	SNP4aa	SNP5 AA	SNP5 Aa	SNP5aa			
Case	0.4	0.3	0.3	0.5	0.3	0.2			
Control	0.33	0.34	0.33	0.45	0.25	0.3			
Genotype	SNP6 AA	SNP6 Aa	SNP6aa	SNP7 AA	SNP7 Aa	SNP7aa	SNP8 AA	SNP8 Aa	SNP8aa
RelativeRisk	3	1	1	2.5	2.5	1	2	1.5	1
Population	0.04	0.32	0.64	0.01	0.18	0.81	0.04	0.32	0.64
EnvFactor1	Level1	Level2	Level3	Level4					
Case	0.1	0.2	0.1	0.6					
Control	0.5	0.2	0.1	0.2					
EnvFactor2	Level1	Level2	Level3	Level4	Level5	Level6			
RelativeRisk	1	1.5	2	1.2	3	2.5			
Population	0.3	0.1	0.1	0.2	0.1	0.2			

Figure 1: An example txt file for **Test Design**

allowed by **Test Build**. Figure 2 gives an example of data for model building (Note that the SNP1 of the second affected individual and the SNP3 of the first unaffected individual are missing, and are coded as -9), which is also freely accessible in the package and online at `https://www.msu.edu/~qlu/software.html`.

5 Using `Bridge`

The main GUI window of `Bridge` is called **Bridge Dialogs** and mainly comprised of five different areas (Figure 3). They are **MenuBar**, **ToolBar**, **Tree area**, **Input area** and **Output area**, which are used for different tasks. In the following, we introduce the functions of five areas in details.

The MenuBar The **MenuBar** has all the options, including loading data, running the analysis, and plotting the ROC curve.

The **File** menu includes several basic functions for R environment settings, including setting working directory, saving and restoring the workspace and exiting the software. The **Test Design** menu is used for designing a risk

Trait	SNP1	SNP2	SNP3	SNP4	SNP5
1	1	1	2	1	1
1	-9	1	1	1	1
1	0	2	1	2	1
...	...	...	...	...	...
0	1	2	-9	2	1
0	0	2	1	2	0
0	1	1	2	1	2
...	...	...	...	...	...

Figure 2: An example txt file for **Test Build**

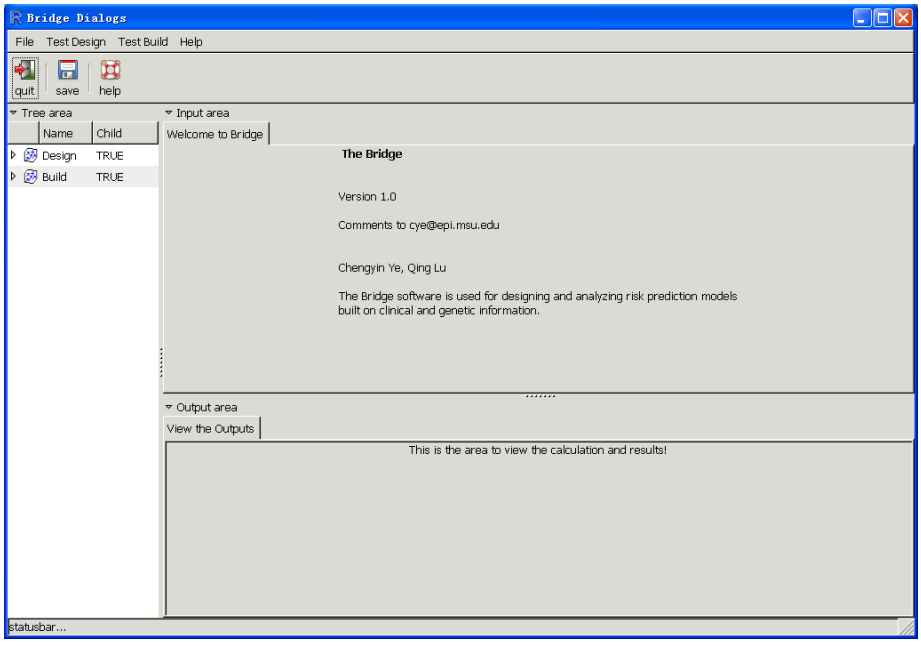

Figure 3: The Dialog window of `Bridge` package

prediction model, while the **Test Build** menu is used to form a risk prediction model. The **Help** menu provides a brief introduction to the **Bridge** package and this vignette file.

The ToolBar includes icons to directly access some functions. The **save** icon is designed to save the results from **Test Design** and **Test Build**, as well as the workspace that has been created. The **Help** icon is used to access the help document.

Tree area provides a tree structure to display data and results. The tree has two roots, **Design** and **Build**, for two correponding sections of the `Bridge` package. By clicking the **arrow** sign, users can view the data or results in the **Input area** or **Output area**.

Input area displays the input data, which allows users to view and modify the entering data. If any tab with the input data has been accidentally closed, it can be reopened by double-clicking the corresponding tab name in the **Tree area**. After all the parameters and data have been entered, the **Run** option can be clicked to run the analysis.

Output area lists all the results from the analysis. For the **Test Build** module, an additional page, named **"The Test Build processing"**, will display the detailed analysis process. Similarly, the tabs of results can be closed and reopened by double-clicking the corresponding tab name in the **Tree area**.

6 Illustration Examples

6.1 An Example of using Test Design

Suppose we have 8 SNPs and 2 environmental risk predictors (Figure 1) which have been reported in previous literatures to be associated with a disease. Based on these 10 predictors, we design a risk prediction model.

6.1.1 Input data

We set the prevalence of disease to be 0.005 by clicking the **Input Disease Prevalence...** option in the **Data Input...** option from the **Test Design** menu. Then, we input the data of genetic and environmental risk predictors manually. We first enter the genetic data by clicking the **Input SNP Data...** option. In the **"Input**

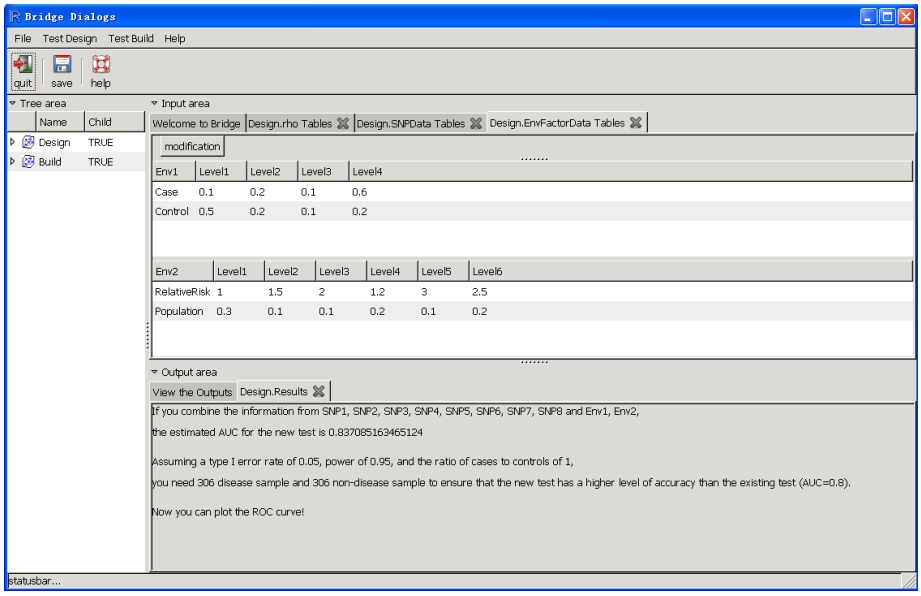

Figure 4: The window of **Test Design** section

the number of SNPs having the following information...” window, we set the SNP numbers of each data type to 2, 1, 2 and 3, respectively, based on this example. Then, in the **“SNP Tables”** window, we enter the data of different types manually. By clicking “OK”, the entering data is listed under **“Design.SNPData Tables”** from the **Input area**. From the table, we can double-check the data and make further modifications (Figure 4). The environmental risk predictors (or genetic risk predictors with more than 3 categories) can be entered by clicking the **Input Environmental Risk Factor Data...** option. In the window of **“Input Environment Risk Factor Number and Level Information...”**, we set the number of risk predictors to 1 for each data type and the levels of the predictors to 4 and 6, respectively. We then enter the data in the **“EnvFactor Tables”** window for each of the environmental risk predictiors. A message of success will be shown if all the data are entered correctly. The final data will be listed in **“Design.EnvFactorData Tables”** for double-checking and modifications (Figure 4).

Alternatively, we can import the data from a txt file. The data file Figure 1 can be imported by using the **Import DataSet...** option from **Data Input...**. The sample data file used in the analysis, DesignData.txt, can be found online at `https://www.msu.edu/~qlu/software.html`.

6.1.2 Run

By clicking the **Run** command in the **Test Design** menu, we allow to choose specific SNPs and environmental risk predictors to design a risk prediction model from the **"Test Design Running..."** window. In this example, we choose all the genetic and environmental risk predictors (Figure 5) for designing a risk prediction model. For sample size calculation, we also set the type-one error to be 0.05, the power to be 0.95, the accuracy level to be 0.8. By clicking the "OK" button, the analysis results are displayed in the **"Design.Results"** window under the the **Output area**. Using all risk predictors, the model reached an estimated AUC value of *0.837*. Totally *306 cases and 306 controls are required to ensure that the new model attained a AUC value higher than AUC=0.8* (Figure 4).

6.1.3 Plot

The ROC curve of the risk prediction model from the design stage can also be displayed by clicking the **Plot ROC Curve** command from the **Test Design** menu, or by double-clicking **Design.Plot** in the **Tree area**. The plot is shown in the `RGui` window and could be saved in various format from the `RGui` (Figure 6).

6.1.4 Save

The results can be saved by choosing the **Save Test Design result** option from the **save** icon in the **ToolBar**. (Figure 9).

6.2 An Example of using Test Build

We use an example to illustrate the use of **Test Build** module (Figure 7). The sample datasets used in the analysis (i.e., Build-Train.txt and Build-Valid.txt) can be found under the data directory of the `Bridge` package, as well as online at `https://www.msu.edu/~qlu/software.html`. The training data (i.e., Build-Train.txt) is used for model building. The data is comprised of 5 SNPs and 3971 individuals, 1491 of which are cases. The validation data (i.e., Build-Valid.txt) is used for model validation. The data has total 1000 individuals, 500 of which are cases.

6.2.1 Import DataSets

The training data can be loaded to the system by clicking the **Import a dataset...** command of the **Data Input...** option from the **Test Build** menu. After choosing the data file in the **"Select a file to import the dataset"** window, we have the

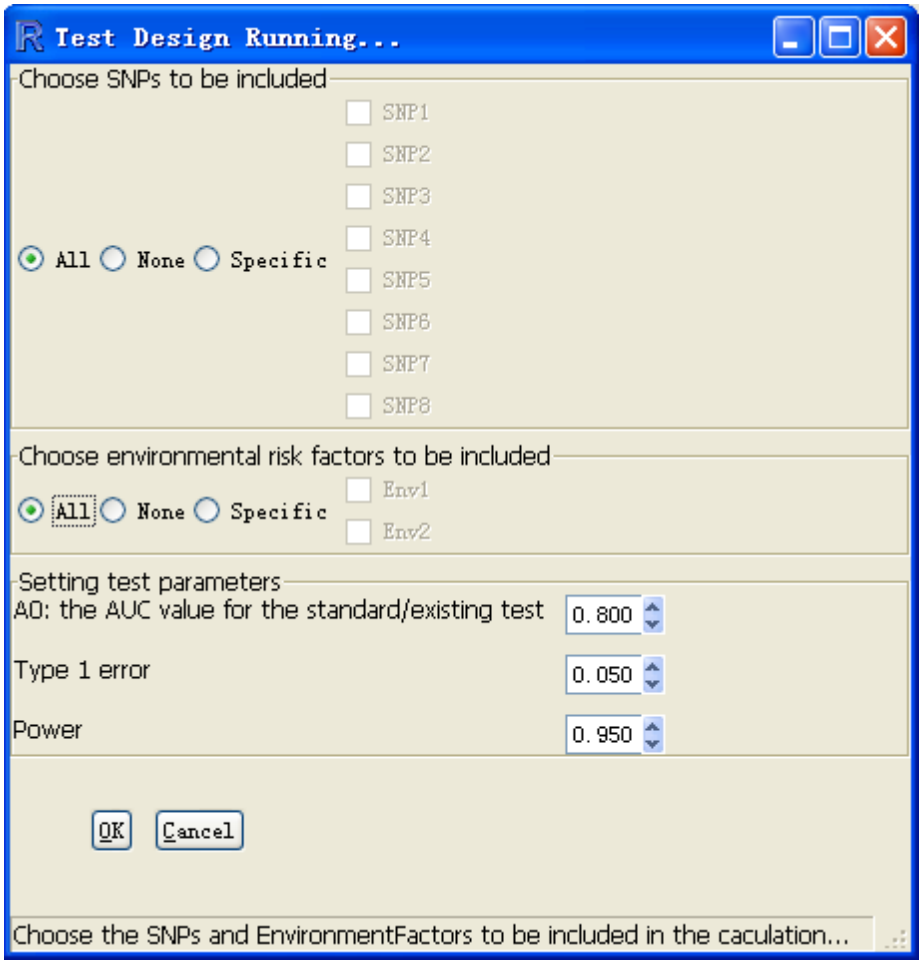

Figure 5: The running window of **Test Design**

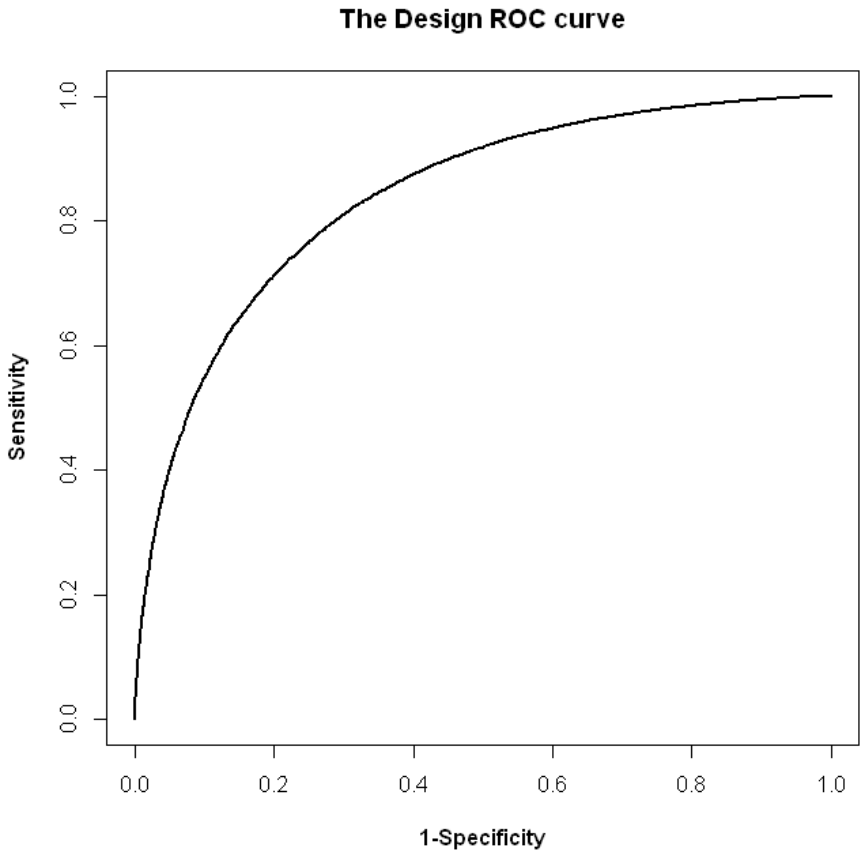

Figure 6: The ROC Curve of the risk prediction model from **Test Design**

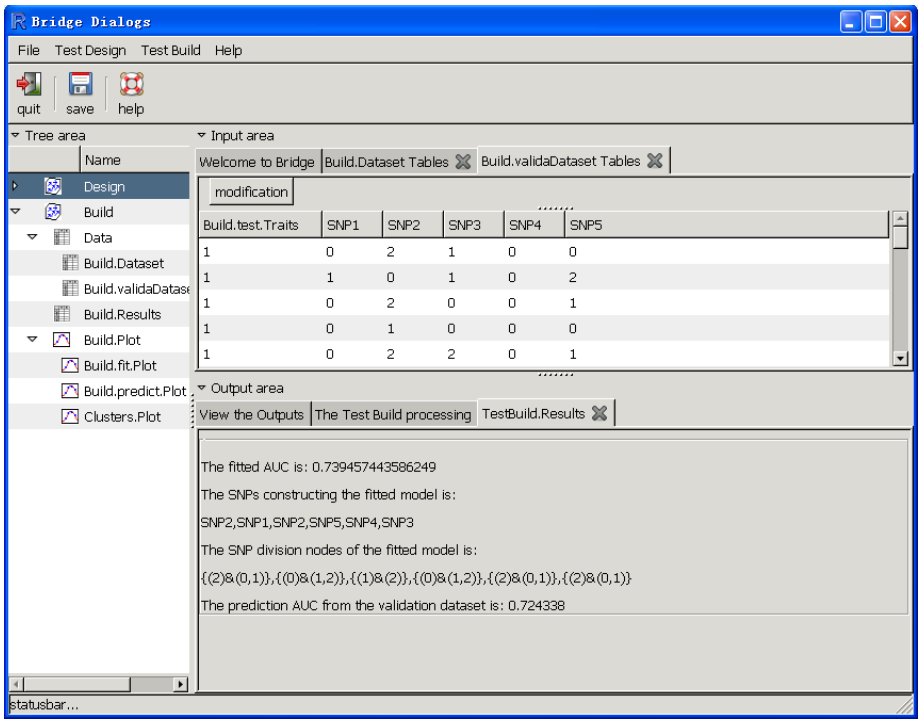

Figure 7: The window of **Test Build** section

option to choose individuals and SNPs to be included in the analysis. For this example, we choose all the SNPs and individuals. The data is then displayed in the **"Build.Dataset Tables"** from the **Input area** for double-checking and further modifications (Figure 7).

Similarly, We also upload the validation data via **"Import a second (validation) dataset..."**. If the validation data is not available, users can also choose the **"Use the previous dataset"** option in the **"Select a file to import the validation dataset"** window, then the analysis will only build a model based on the training dataset. The validation data is available under **"Build.validaDataset Tables"** in the **Input area**. Again, all of data could also be reviewed from the **Tree area** (Figure 7).

6.2.2 Run

By clicking the **Run** command under the **Test Build** menu, users will be asked for addtional information for the anaysis. **The number of folds of the cross-validation** were used in the process of cross validation to determine the model complexity. By default, 10-fold cross-validation is used. However, user can also choose a small number to speed the analysis. **The maximum AUC value** specifies the maximum value of AUC for the model. By defaulty, AUC value of 1 is used. However, user can choose a reasonable AUC value (0.5-1) to speed the analysis. **An option to control the way of clustering genotype groups** parameter allows users to decide possible ways of clustering risk groups. If 3 is selected, all three possible ways of clustering risk groups, i.e., (2)VS(0,1), (0)VS(2,1) and (1)VS(0,2), will be considered, while only the first two ways of clustering will be considered if 2 is chosen. In this example, we choose the number of folds for the cross-validation to be 10, **The maximum AUC value** to be 1, the **An option to control the way of clustering genotype groups** to be 3. **"The Test Build processing"** in the **Output area** (Figure 7) displays the detailed steps of the analysis. When the analysis is completed, the result is shown in the **"TestBuild.Results"** from the **Output area**. Based on the given data, the fitted AUC from our anlaysis is *0.739* and the prediction AUC is *0.724*.

6.2.3 Plot

Similarly, we can plot the ROC curve for the formed model. The fitted ROC curve and the prediction ROC curve can be generated by clicking the **Plot ROC Curve** command in the **Test Build** menu. Both plots can be saved from `RGui` (Figure 8).

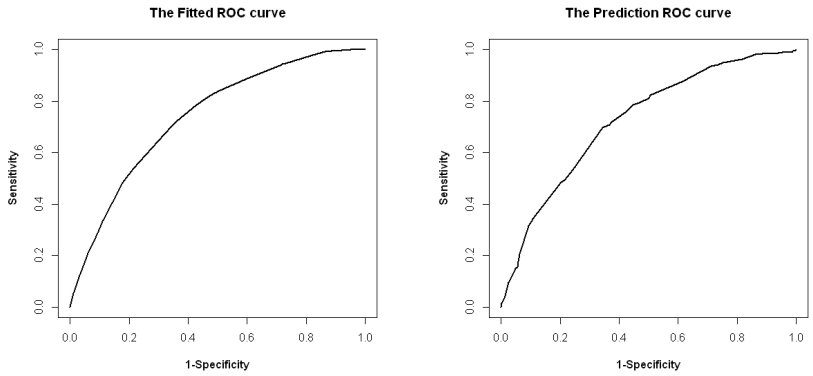

Figure 8: The optimal ROC Curves of **Test Build** results

6.2.4 Save

We click the **Save Test Build result** option from the **save** icon in the **ToolBar** to save the **"TestBuild.Results"** (Figure 9).

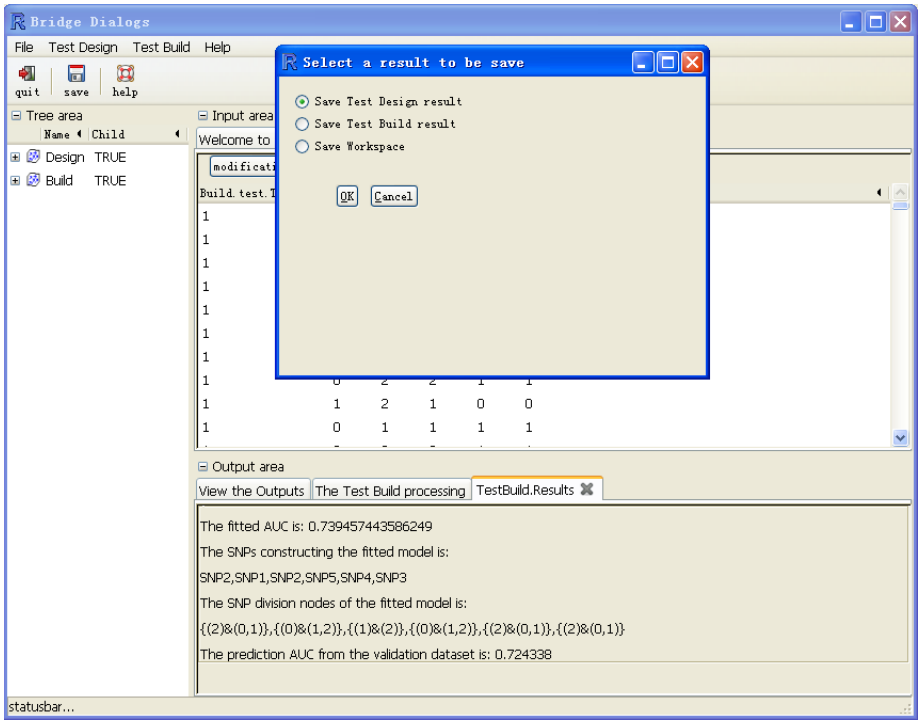

Figure 9: The window of **Test Build** section