Model Selection Tool

This tutorial provides information about Partek model selection tool, how to use this function and some common mistakes which we should avoid to do. The dataset used in the tutorial is a simulated human microarray intensity values in log space. The data is not used for diagnostic procedure, only to show how to use the function.

Select a Classification Model

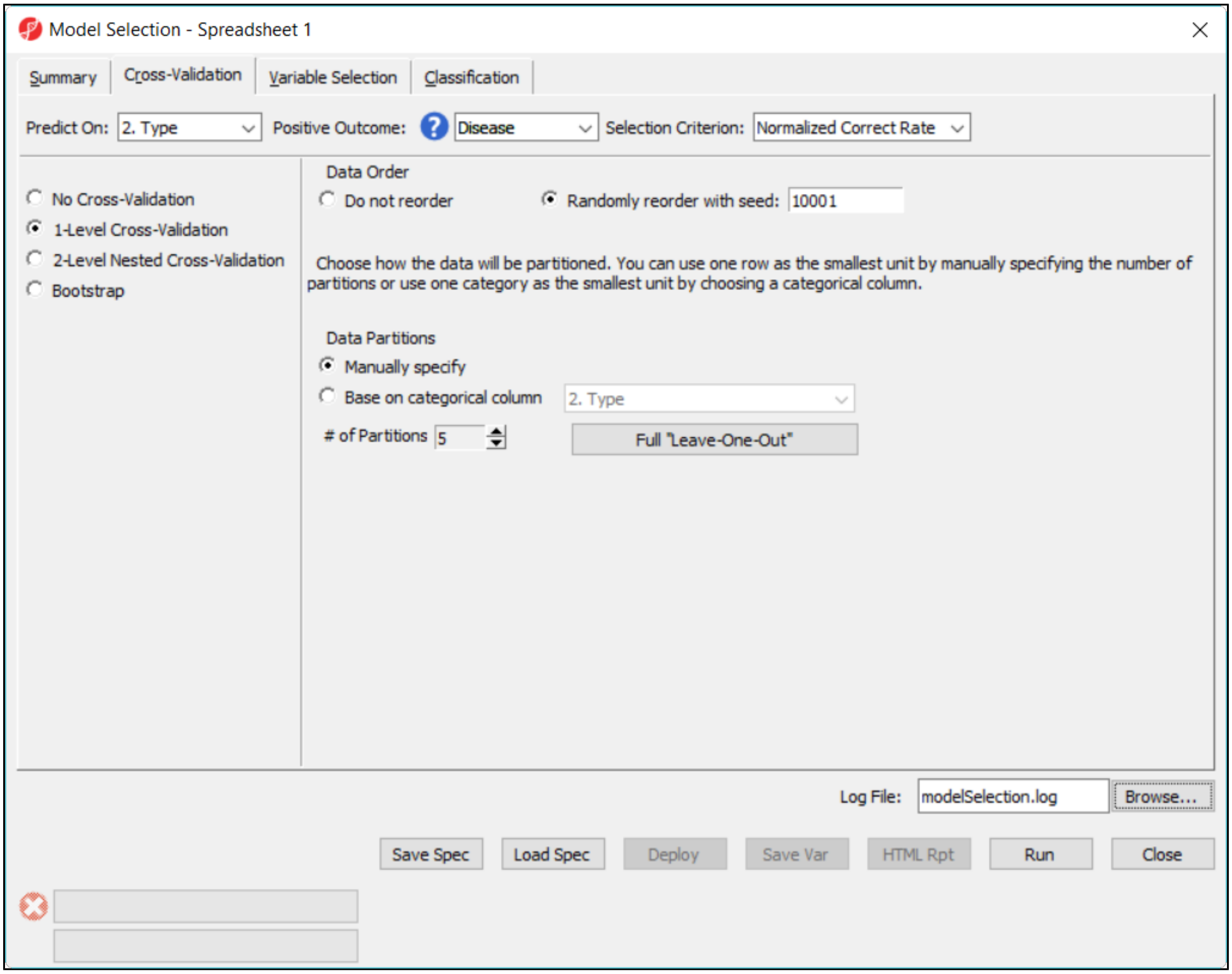
Download the zip file from here. The download contains the following files:

- Training set data: 28 samples (11 disease samples and 15 normal samples) on 9953 genes
- Test set data: 8 samples on 9953 genes
- configuration of the model builder (.pcms file)
- 36 sample data set – total of training and test samples
- deployed model (.pbb file)

A classification model has two parts: variables and classifier. The model selection tool in Partek Genomics Suite uses cross-validation to choose the best classification model and gives the accuracy estimate of the best model.

1-level cross-validation is used to select the best model to deploy. There are two ways to report the unbiased accuracy estimate (or correct rate): 2-level cross validation on the same data set, or deploy the model on a independent test set. We will show both in this tutorial.

- Open Partek Genomics Suite, choose **File > Open...** from the main menu to open the **Training.fmt**
- Select **Tools > Predict > Model Selection** from the Partek main menu
- In *Cross-Validation* tab, choose to *Predict on* **Type**, *Positive Outcome* is **Disease**, *Selection Criterion* is **Normalized Correct Rate** (Figure 1)
- Choose **1-Level Cross-Validation option**, and use *Manually specify* partition option as **5.** The idea of 1-level cross validation option is to select the best model to deploy on the test data set.



Model Selection - Spreadsheet 1

Predict On: 2. Type

Positive Outcome: Disease

Selection Criterion: Normalized Correct Rate

○ No Cross-Validation
● 1-Level Cross-Validation
○ 2-Level Nested Cross-Validation
○ Bootstrap

Data Order

○ Do not reorder ● Randomly reorder with seed: 10001

Choose how the data will be partitioned. You can use one row as the smallest unit by manually specifying the number of partitions or use one category as the smallest unit by choosing a categorical column.

Data Partitions

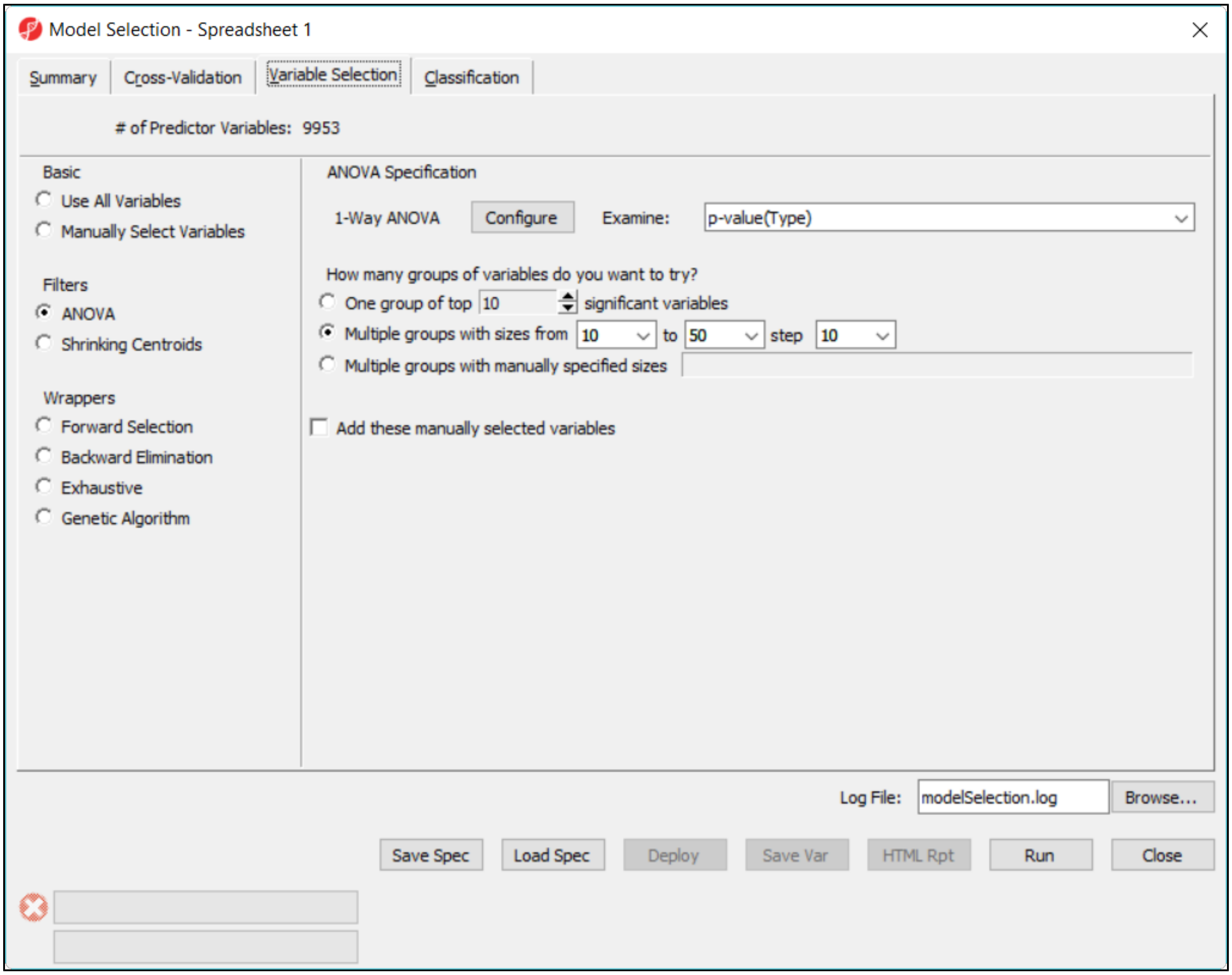
● Manually specify
○ Base on categorical column 2. Type

# of Partitions 5

Log File: modelSelection.log

Figure 13. Model selection dialog: 1-level cross validation configuration

- Choose **Variable Selection** tab, to use ANOVA to select variables. The number of genes selected are based on the p-value generated from the 1-way ANOVA model which factor is *Type*. In each iteration of cross validation, we will use the training set to perform ANOVA, take the top N number of genes with the most significant p-values to build the classifier. The *Configure* button allow you to specify ANOVA model if you want to include multiple factors (Figure 2).
- Since we don't know how many genes should be used to build the model, we will try to use 10, 20, 30, 40, 50 genes – the more options you try, the longer time it takes to run. In the *How many groups of variables do you want to try*, select **Multiple groups with sizes from 10 to 50 step 10**


Model Selection - Spreadsheet 1

# of Predictor Variables: 9953

Basic
○ Use All Variables
○ Manually Select Variables

Filters
● ANOVA
○ Shrinking Centroids

Wrappers
○ Forward Selection
○ Backward Elimination
○ Exhaustive
○ Genetic Algorithm

ANOVA Specification

1-Way ANOVA Configure Examine: p-value(Type)

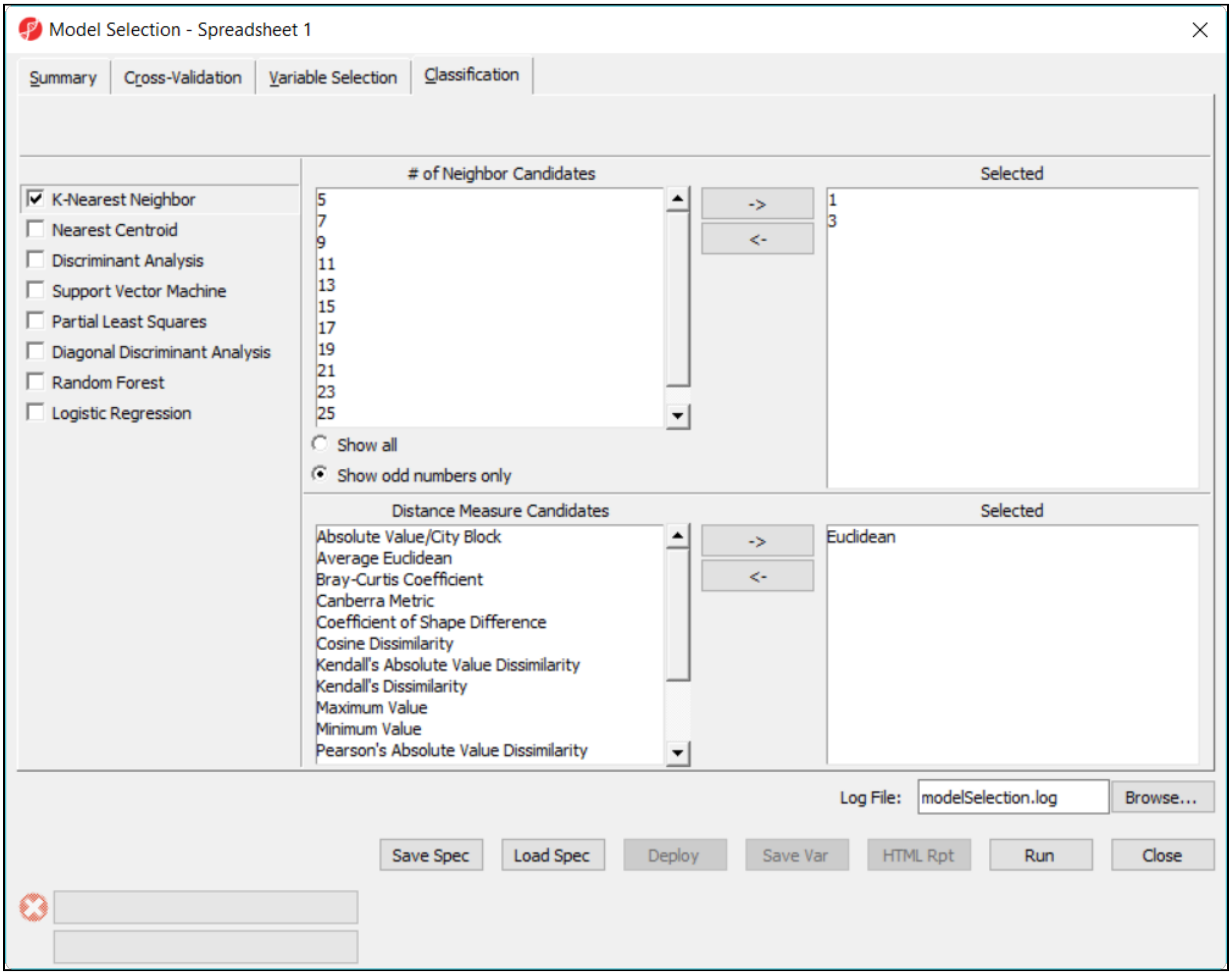
How many groups of variables do you want to try?
○ One group of top 10 significant variables
● Multiple groups with sizes from 10 to 50 step 10
○ Multiple groups with manually specified sizes

□ Add these manually selected variables

Log File: modelSelection.log

Figure 14. Model selection dialog: Variable selection configuration

- Click on **Classification** tab, select **K-Nearest Neighbor**, choose **1** and **3** neighbors using default **Euclidean** distance measure (Figure 3)



☑ K-Nearest Neighbor
□ Nearest Centroid
□ Discriminant Analysis
□ Support Vector Machine
□ Partial Least Squares
□ Diagonal Discriminant Analysis
□ Random Forest
□ Logistic Regression

# of Neighbor Candidates: 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25

Selected: 1, 3

○ Show all
● Show odd numbers only

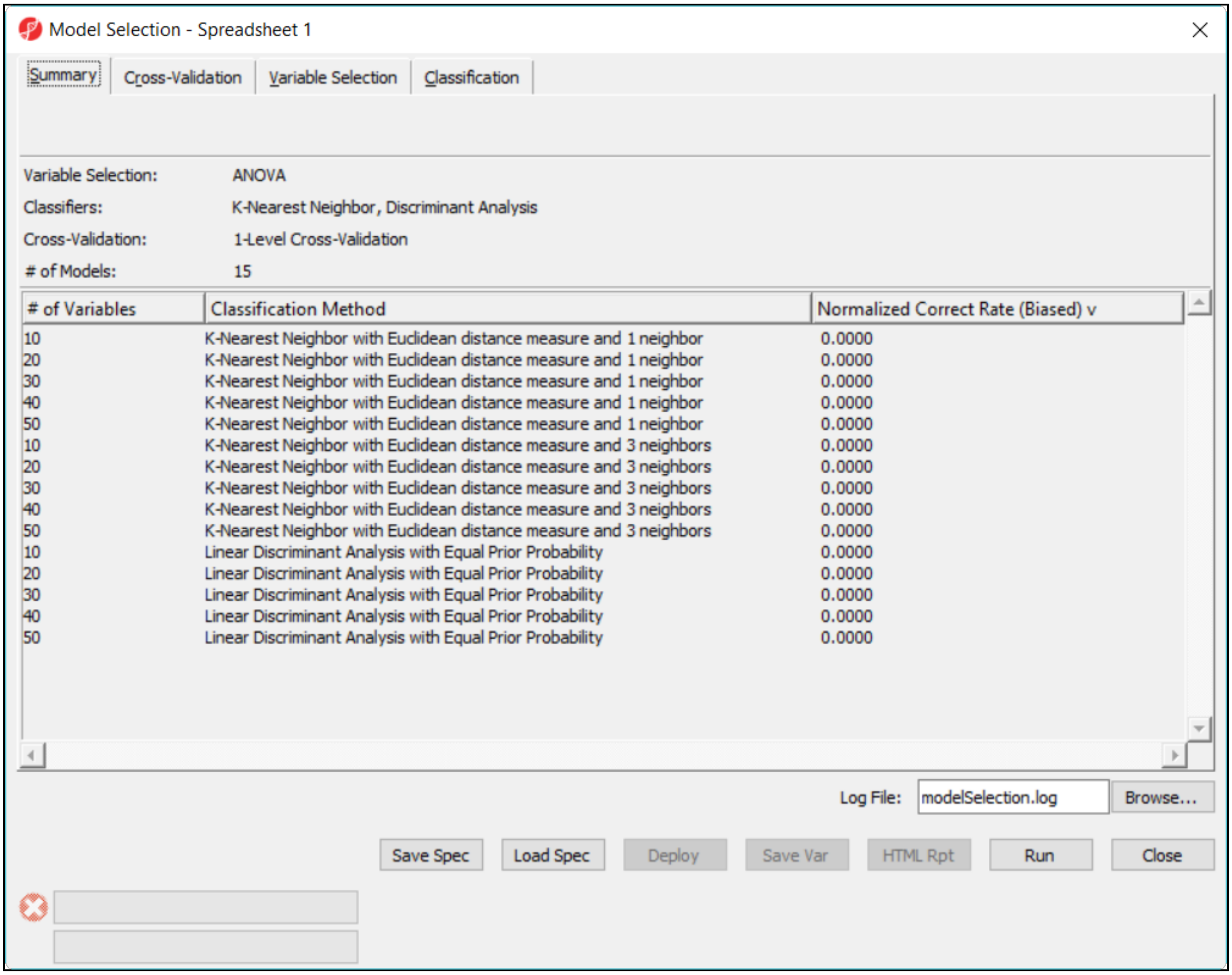
Distance Measure Candidates: Absolute Value/City Block, Average Euclidean, Bray-Curtis Coefficient, Canberra Metric, Coefficient of Shape Difference, Cosine Dissimilarity, Kendall's Absolute Value Dissimilarity, Kendall's Dissimilarity, Maximum Value, Minimum Value, Pearson's Absolute Value Dissimilarity

Selected: Euclidean

Log File: modelSelection.log

Figure 15. Model selection dialog: K-nearest neighbor configuration

- Select **Discriminant Analysis** option, use the default setting which has the *Linear with equal prior probabilities* option checked
- Click on **Summary** tab, we have configured 15 models to choose from (Figure 3)



Variable Selection: ANOVA

Classifiers: K-Nearest Neighbor, Discriminant Analysis

Cross-Validation: 1-Level Cross-Validation

# of Models: 15

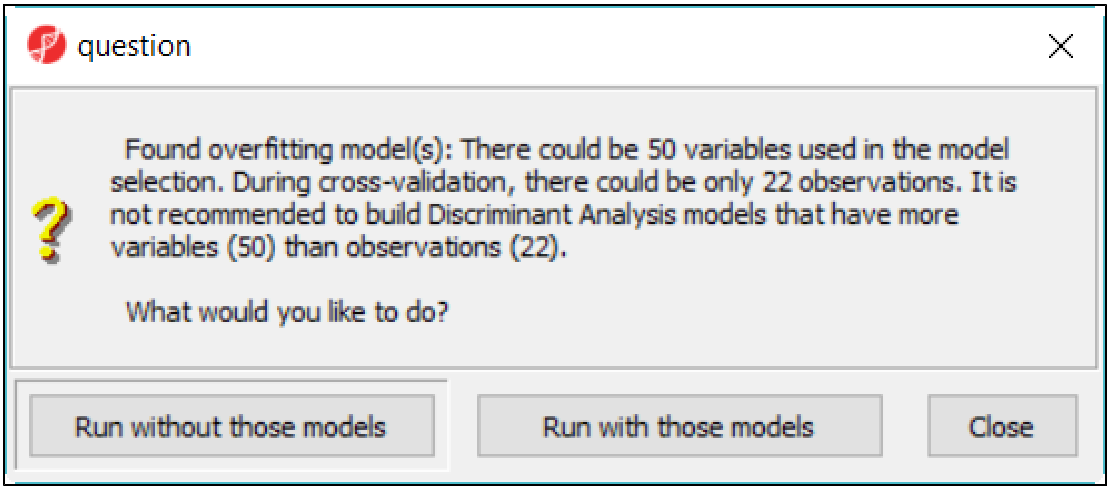
# of Variables	Classification Method	Normalized Correct Rate (Biased) v
10	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	0.0000
20	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	0.0000
30	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	0.0000
40	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	0.0000
50	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	0.0000
10	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	0.0000
20	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	0.0000
30	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	0.0000
40	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	0.0000
50	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	0.0000
10	Linear Discriminant Analysis with Equal Prior Probability	0.0000
20	Linear Discriminant Analysis with Equal Prior Probability	0.0000
30	Linear Discriminant Analysis with Equal Prior Probability	0.0000
40	Linear Discriminant Analysis with Equal Prior Probability	0.0000
50	Linear Discriminant Analysis with Equal Prior Probability	0.0000

Log File: modelSelection.log

Figure 16. Model selection dialog: Summary page

The more models configured, the long time it takes to run, in this example, in order to save time, we only specified 15 models and choose 5-fold cross-validation. You can also click on **Load Spec** button to load the above configuration from file **tutorial.pcms**

When you click on **Run**, a dialog as the one in Figure 4 will be displayed, notifying you that some classifiers, like discriminant analysis, are not recommended on dataset with more variables than samples.



question

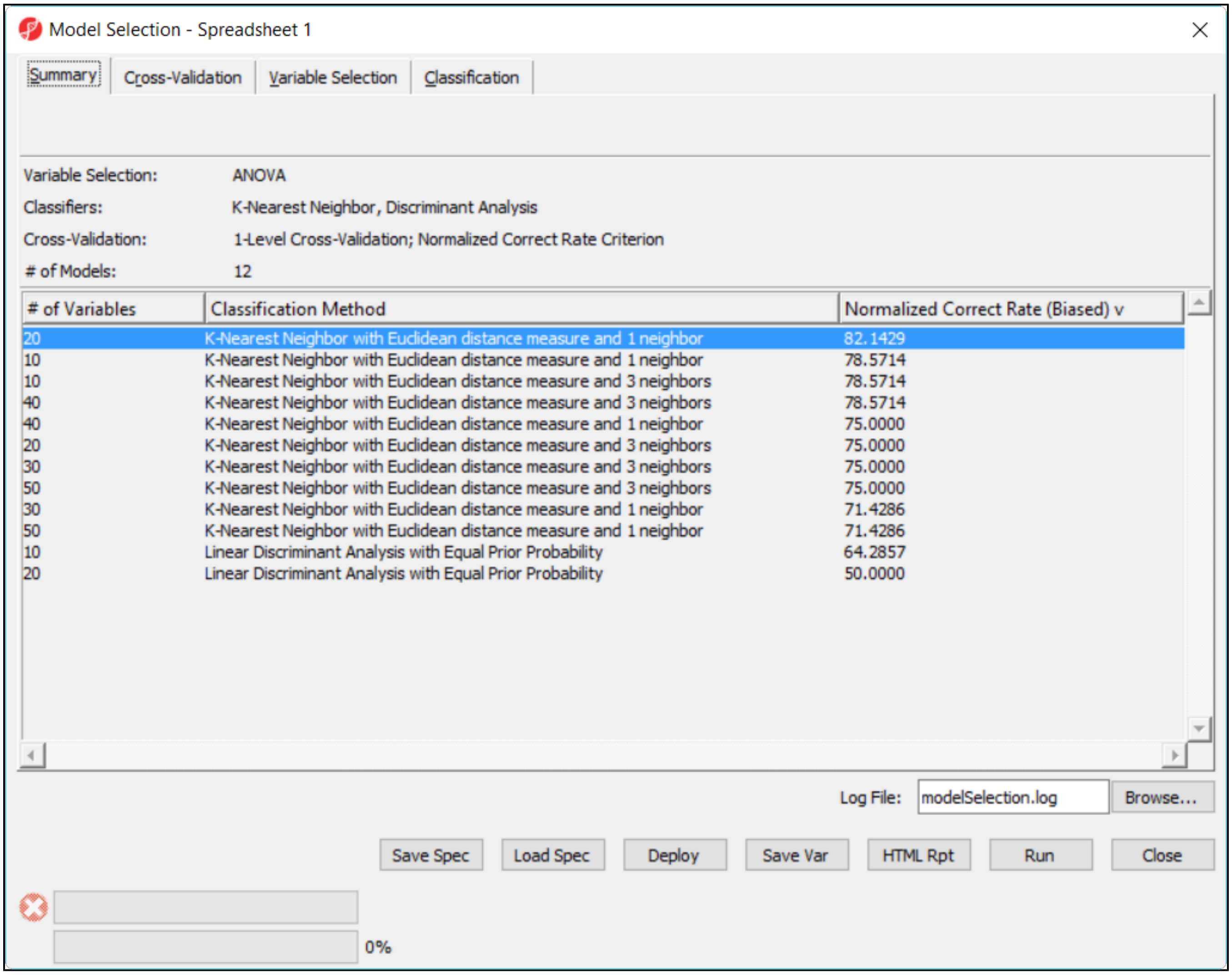
Found overfitting model(s): There could be 50 variables used in the model selection. During cross-validation, there could be only 22 observations. It is not recommended to build Discriminant Analysis models that have more variables (50) than observations (22).

What would you like to do?

Figure 17. A notification that discriminant analysis model is not recommended on data with more variables than samples

- Click **Run without those models** button to dismiss the dialog, leaving12 models in this model space

Since we are doing 5-fold cross validation, there will be 6 samples held out as test set in each iteration, and the models are built on the remaining 22 samples training set. After it is done, all the 12 models have been tested on the 28 samples, and the correct rate will reported, they are displayed in the summary page in descending order of the normalized biased ⚠ correct rate, the top one is the best model among the 12 models (Figure 5).



Variable Selection: ANOVA

Classifiers: K-Nearest Neighbor, Discriminant Analysis

Cross-Validation: 1-Level Cross-Validation; Normalized Correct Rate Criterion

# of Models: 12

# of Variables	Classification Method	Normalized Correct Rate (Biased) v
20	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	82.1429
10	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	78.5714
10	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	78.5714
40	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	78.5714
40	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	75.0000
20	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
30	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
50	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
30	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	71.4286
50	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	71.4286
10	Linear Discriminant Analysis with Equal Prior Probability	64.2857
20	Linear Discriminant Analysis with Equal Prior Probability	50.0000

Log File: modelSelection.log

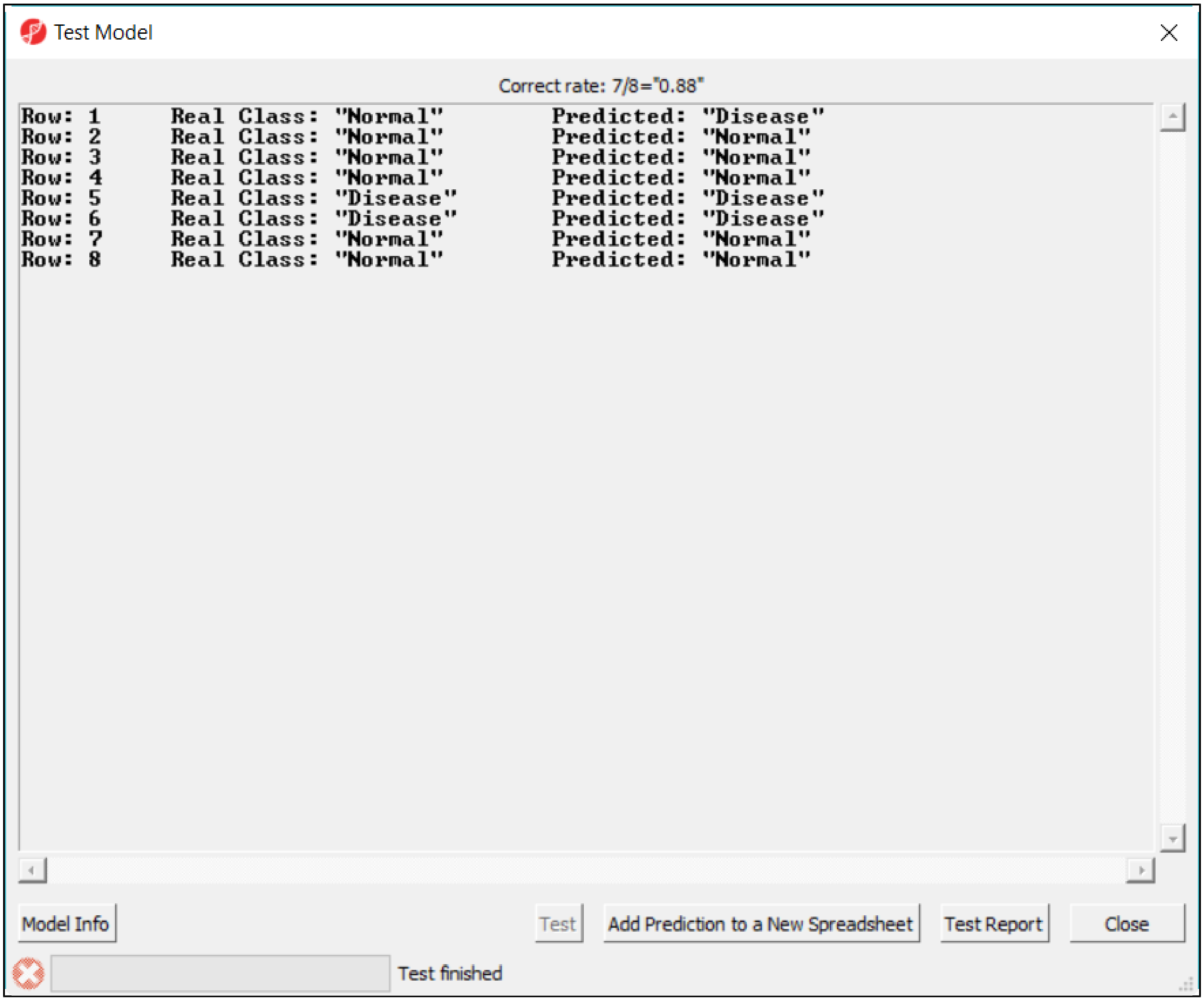
Figure 18. One-level cross-validation result: 20 variables 3 nearest neighbor with Euclidean distance measure is the best model among the 12 models on the 28 sample data

- Click on **Deploy** button to leaunch the model using the whole dataset, but first save the file as **20var-1NN-Euclidean.ppb**. It will run ANOVA on the 28 samples to generate the top 20 genes and build a model using 3 K-Nearest neighbor based on Euclidean distance measure.
- Since the deployed model was from the whole 28 samples, in order to know the correct rate, we need a test set to run the model on.

Deploying a Model

To get unbiased correct rate, the test set sample must be independent from the training set. Now we are going to load another dataset, it has 8 samples with logged intensity values on the set of genes as that of the training data set. To use a complete independent test set to get correct rate is called hold-out validation.

- Choose **File > Open...** to browse and open **testSet.fmt**
- Choose T**ools > Predict > Run Deployed Model...** from the menu
- Select **20var-3NN-Euclidean.ppb** to open, click on **Test** button to run, the C*orrect rate* (= accuracy) is reported on the top of the dialog (Figure 6)



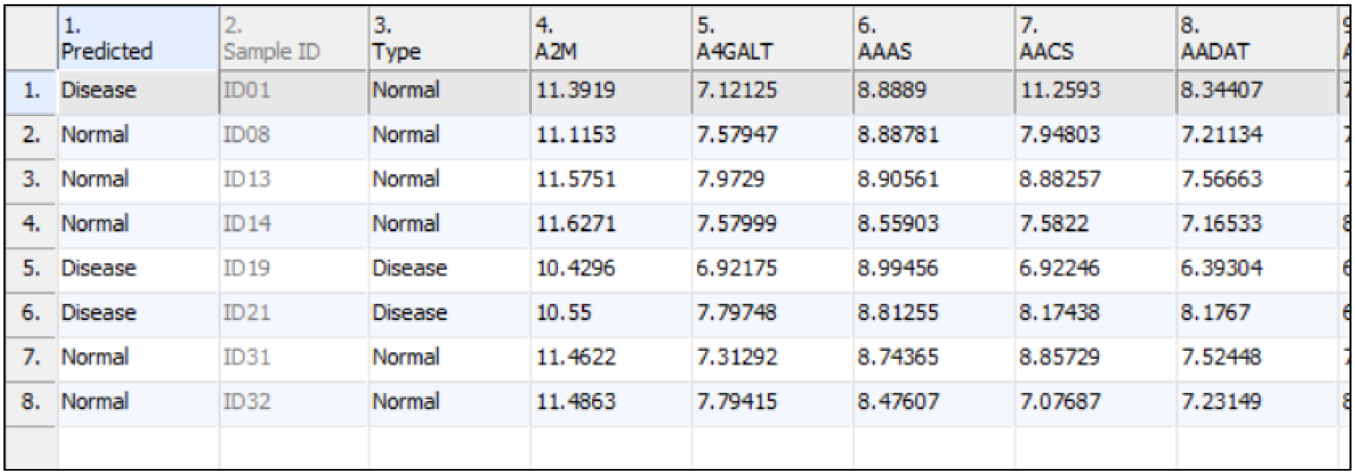
Correct rate: 7/8="0.88"

```
Row: 1     Real Class: "Normal"         Predicted: "Disease"
Row: 2     Real Class: "Normal"         Predicted: "Normal"
Row: 3     Real Class: "Normal"         Predicted: "Normal"
Row: 4     Real Class: "Normal"         Predicted: "Normal"
Row: 5     Real Class: "Disease"        Predicted: "Disease"
Row: 6     Real Class: "Disease"        Predicted: "Disease"
Row: 7     Real Class: "Normal"         Predicted: "Normal"
Row: 8     Real Class: "Normal"         Predicted: "Normal"
```

Test finished

Figure 19. Report on deploying a model on a test data set

- Click **Add Prediction to New Spreadsheet** to generate new spreadsheet with a predicted class name in the first column, the samples (rows) whose predicted and true class name are different are highlighted (Figure 7)



	1. Predicted	2. Sample ID	3. Type	4. A2M	5. A4GALT	6. AAAS	7. AACS	8. AADAT
1.	Disease	ID01	Normal	11.3919	7.12125	8.8889	11.2593	8.34407
2.	Normal	ID08	Normal	11.1153	7.57947	8.88781	7.94803	7.21134
3.	Normal	ID13	Normal	11.5751	7.9729	8.90561	8.88257	7.56663
4.	Normal	ID14	Normal	11.6271	7.57999	8.55903	7.5822	7.16533
5.	Disease	ID19	Disease	10.4296	6.92175	8.99456	6.92246	6.39304
6.	Disease	ID21	Disease	10.55	7.79748	8.81255	8.17438	8.1767
7.	Normal	ID31	Normal	11.4622	7.31292	8.74365	8.85729	7.52448
8.	Normal	ID32	Normal	11.4863	7.79415	8.47607	7.07687	7.23149

Figure 20. Test deployed model on test set report on spreadsheet

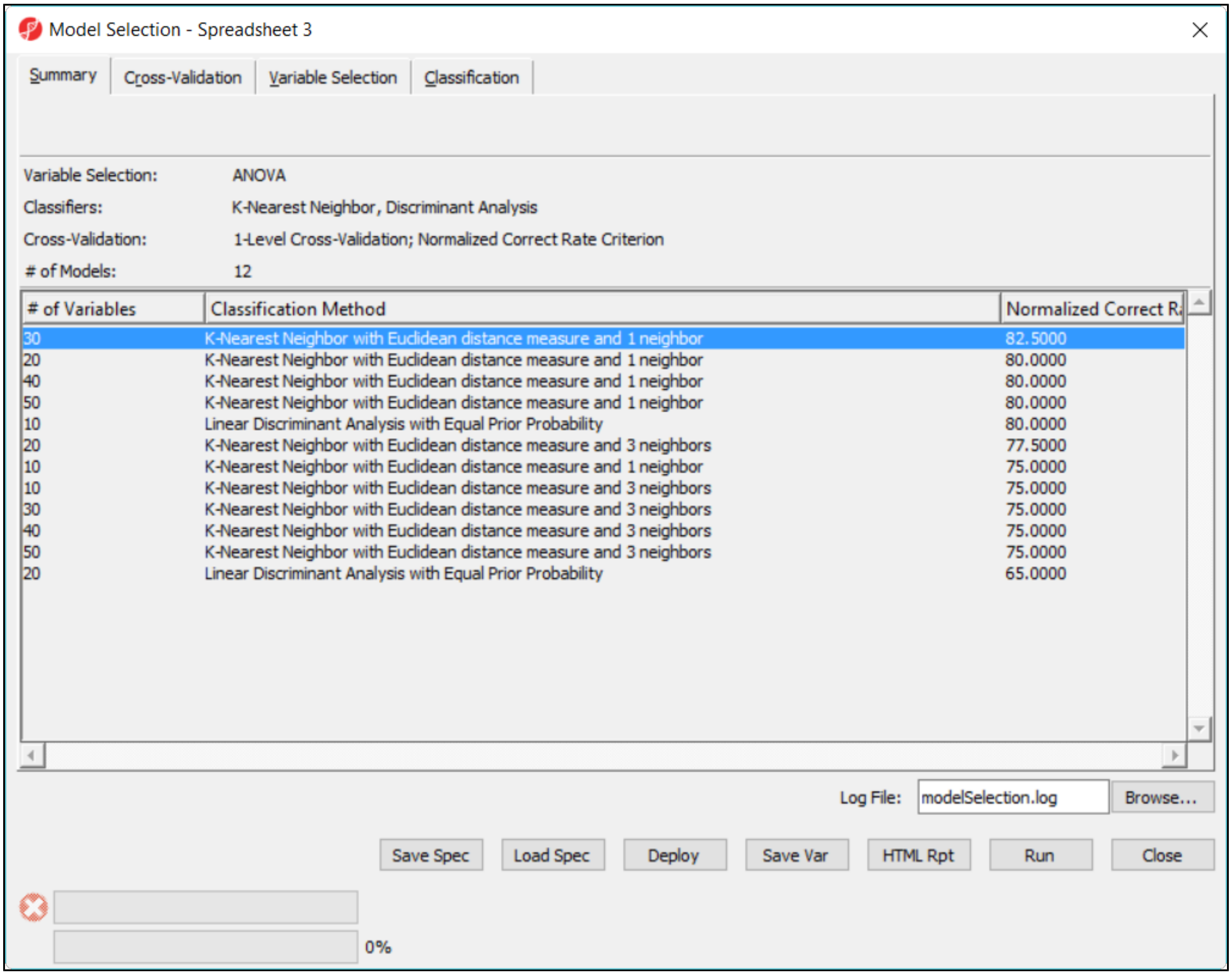
- Click on **Test Report** will generate a report in HTML format
- Click **Close** to dismiss the dialog

Hold-out validation have to split the whole data into two parts: training set and test set. However, genomic data (like microarray or NGS data) typically doesn't contain a large number of samples, os using hold-out method, we have to make the training and test test even smaller. When the sample size is small (here the example data is just illustrate the function), the result is not precise. As a rule of a thumb, you should have at least 100 test samples to properly measure the correct rate with a useful precision. In the other words, the larger the size of training set, the better efficiency of the fitted predicted models are; the bigger size of test set, the better power of validation.

Another method to get unbiased accuracy estimate is to perform a 2-level cross validation on all the available samples (here: utilize the 36 samples set), so thta you don't have to split the data. The following steps show how to use all the 36 samples to select the best model and get the accuracy estimate.

- Choose **File > Open...** to browse to and open the file **36samples.fmt**
- Choose T**ools > Predict > Model Selection...** from the menu
- Click on **Load Spec** to select **tutorial.pcms**
- Click **Run** on 1-level cross validation to select the best model using 36 samples

The best model is 30 variables using 1-Nearest Neighbor with Euclidean distance measure (Figure 8).



Variable Selection: ANOVA

Classifiers: K-Nearest Neighbor, Discriminant Analysis

Cross-Validation: 1-Level Cross-Validation; Normalized Correct Rate Criterion

# of Models: 12

# of Variables	Classification Method	Normalized Correct R
30	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	82.5000
20	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	80.0000
40	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	80.0000
50	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	80.0000
10	Linear Discriminant Analysis with Equal Prior Probability	80.0000
20	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	77.5000
10	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	75.0000
10	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
30	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
40	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
50	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors	75.0000
20	Linear Discriminant Analysis with Equal Prior Probability	65.0000

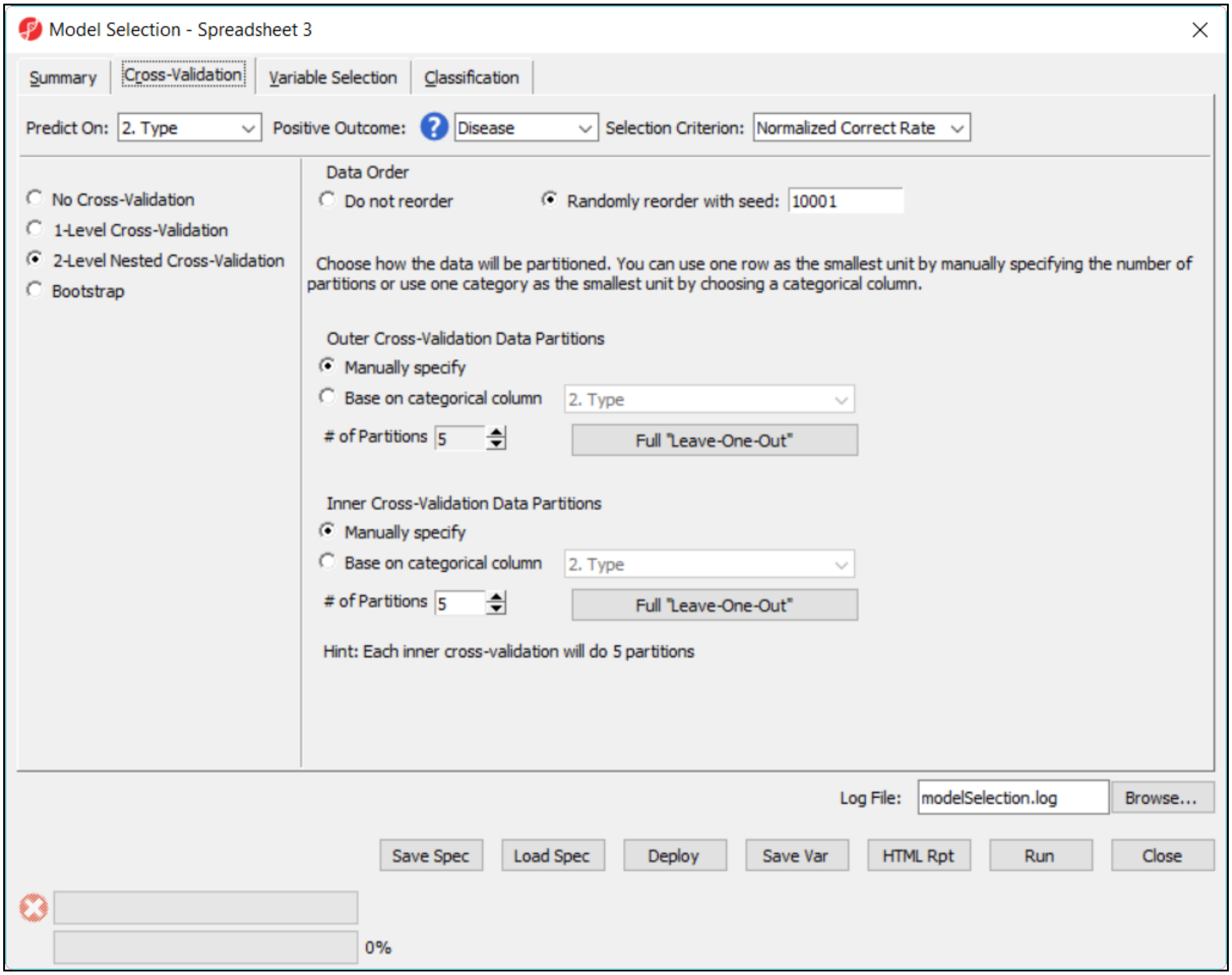
Log File: modelSelection.log

Figure 21. One-level cross-validation result: 30 variables 1 nearest neighbor with Euclidean distance measure is the best model among the 12 models on the 36 sample data

- Click on the model with best correct rate and deploy the model

Since there is no separate data to test the correct rate of the best model in the 12 model space, we will do a 2-level cross-validation to get the accuracy estimate.

- Click on **Cross-Validation** tab, choose *2-Level Nested Cross-Validation*, specify the *number of Partition* as **5** for both, level everything else the same and click **Run** (Figure 9)



Model Selection - Spreadsheet 3

Predict On: 2. Type   Positive Outcome: Disease   Selection Criterion: Normalized Correct Rate

○ No Cross-Validation
○ 1-Level Cross-Validation
● 2-Level Nested Cross-Validation
○ Bootstrap

Data Order
○ Do not reorder   ● Randomly reorder with seed: 10001

Choose how the data will be partitioned. You can use one row as the smallest unit by manually specifying the number of partitions or use one category as the smallest unit by choosing a categorical column.

Outer Cross-Validation Data Partitions
● Manually specify
○ Base on categorical column 2. Type
# of Partitions 5

Inner Cross-Validation Data Partitions
● Manually specify
○ Base on categorical column 2. Type
# of Partitions 5

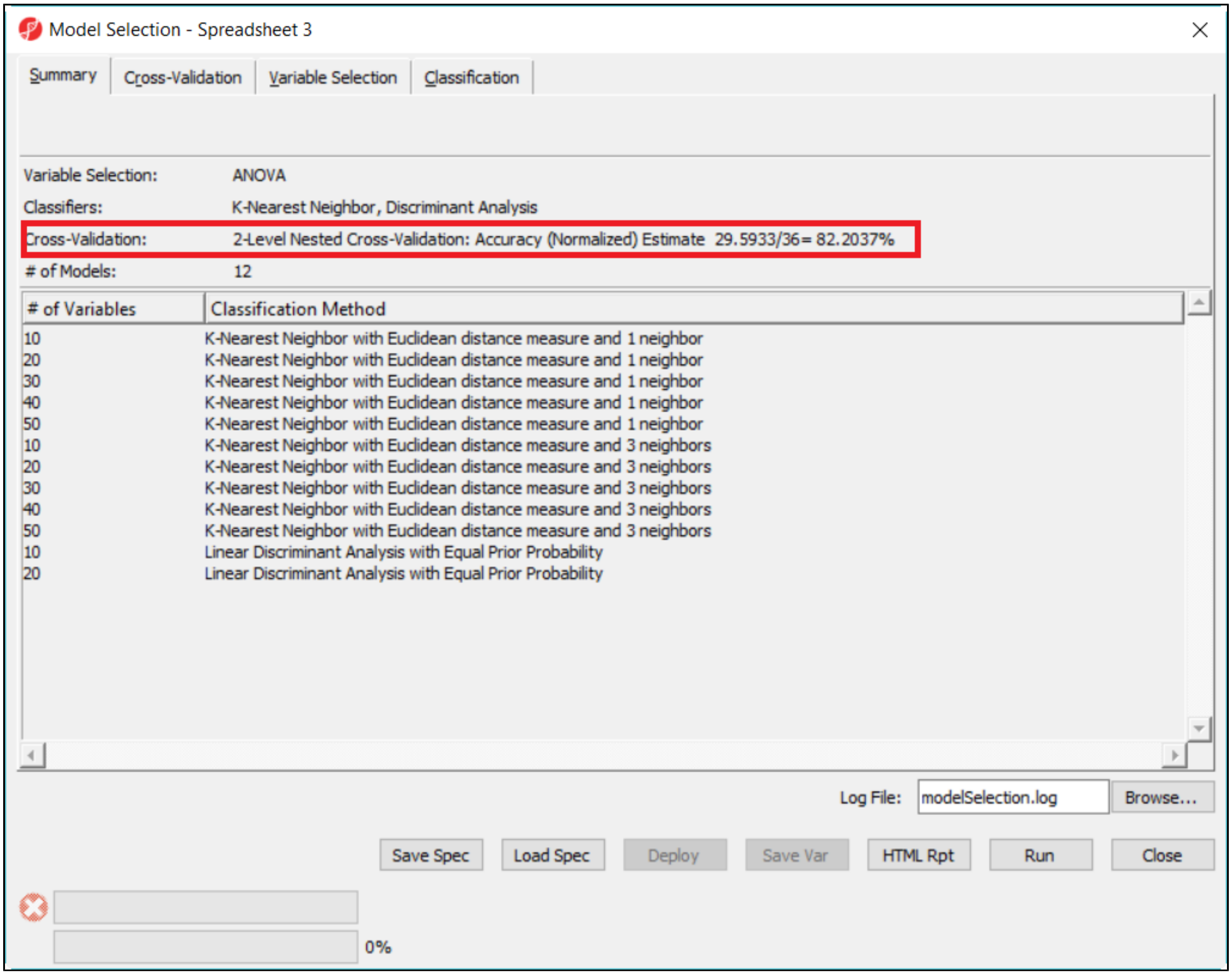
Hint: Each inner cross-validation will do 5 partitions

Log File: modelSelection.log

0%

Figure 22. Two-level cross-validation configuration setup

After it is done, you will get a report like the one in Figure 10. The highligted number is the unbiased accuracy estimate of the best model in the 12 model space.



Variable Selection: ANOVA

Classifiers: K-Nearest Neighbor, Discriminant Analysis

Cross-Validation: 2-Level Nested Cross-Validation: Accuracy (Normalized) Estimate 29.5933/36= 82.2037%

# of Models: 12

# of Variables	Classification Method
10	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor
20	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor
30	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor
40	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor
50	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor
10	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors
20	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors
30	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors
40	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors
50	K-Nearest Neighbor with Euclidean distance measure and 3 neighbors
10	Linear Discriminant Analysis with Equal Prior Probability
20	Linear Discriminant Analysis with Equal Prior Probability

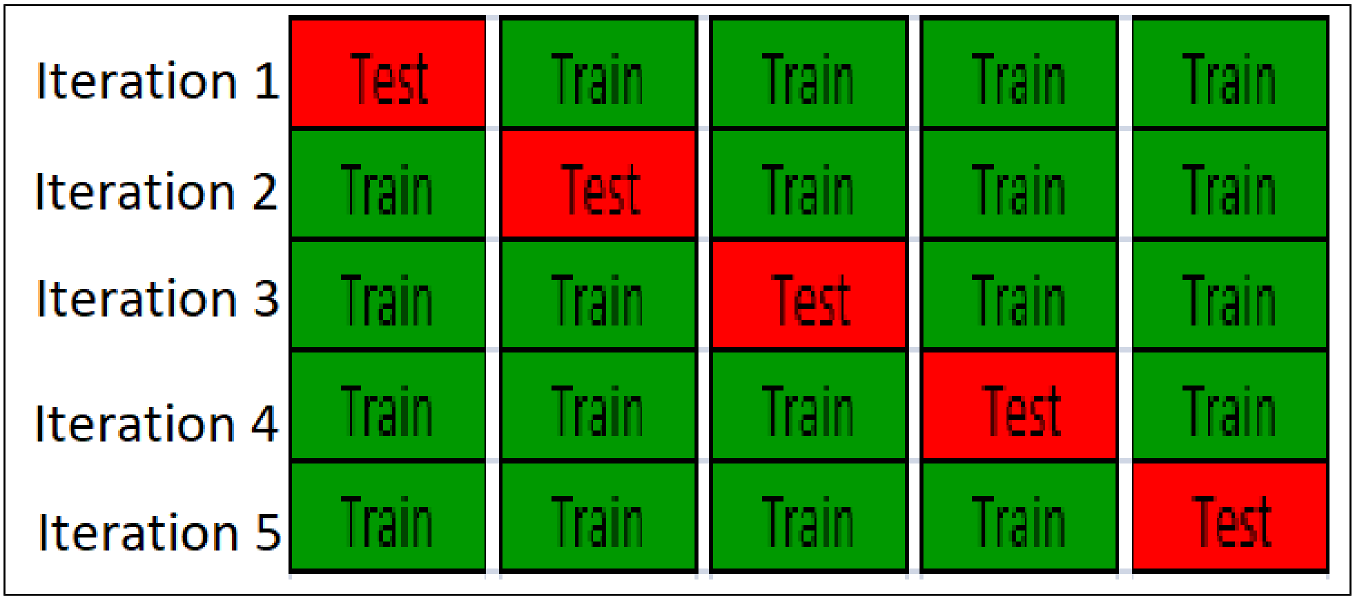
Log File: modelSelection.log

Figure 23. Two level cross-validation report. The highlighted model had the highest accuracy

Cross-validation

Cross validation is used to esimate the accuracy of the predictive model, it is the solution to overfitting problem. One example of ovefitting is testing the model on the same data set when the model is trained. It is like to give sample questions to students to practice before exam, and the exact same questions are given to the students during exam, so the result is biased.

In cross-validation, the data is partition the data into training set and test set, build a model on training set and validate the model on the test set, so test set is completely independing from model traininig. An example of K-fold cross-validation is to randomly divide the whole data into k equal sized subsets, take one subset as test set, use the remanining k-1 subset to training a model, it will repeat K times, so all the samples are used for both training and test, and each sample is tested once. The following figure is showing 5-fold cross-validation:



Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Figure 24. 5-fold cross-validation

Common Mistakes

In Partek model selection tool, the cross-validation is performed first. Each iteration of cross-valiation, the variable selection and classification are performed on the training set, and the test set is completely independent to validate the model. One common mistake is to select variable beforehand, e.g. using perform ANOVA on the whole dataset and use ANOVA's result to select top genes, and perform the cross-valiation to get correct rate. In this case, the test sets in cross validation were used in the variable selection, it is not independend from the training set, so the result will be biased.

Another common mistake is to run 1-level cross-validation with multiple models, and report the correct rate of the best model as the estimate of generalization correct rate, This correct rate is optimistically biased. The reason is that in 1-level cross validation, the test set is used to select the best model, the test set is not independent anymore in terms of estimating correct rate on a unseen dataset. So either use 2-level cross-validation option or use another independ set to get the accuracy estimate, the idea here is to partition the data into 3 sets: training set, validation set and test set. Train the models on the training set, validation set is used to select the best model, and test set is used to generate an unbiased accuracy estimate.

Additional Assistance

If you need additional assistance, please visit our support page to submit a help ticket or find phone numbers for regional support.

Your Rating: ☆☆☆☆☆ Results: ★★★★⯪ 39 rates