Model Selection

Introduction

This user guide provides information about the Partek Model Selection tool. Topics covered include using a Down syndrome data set to demonstrate the usage of the Partek Model Selection tool, introducing the concept of cross-validation, and giving two common mistake examples to look for when doing model selection and classification.

Model Selection

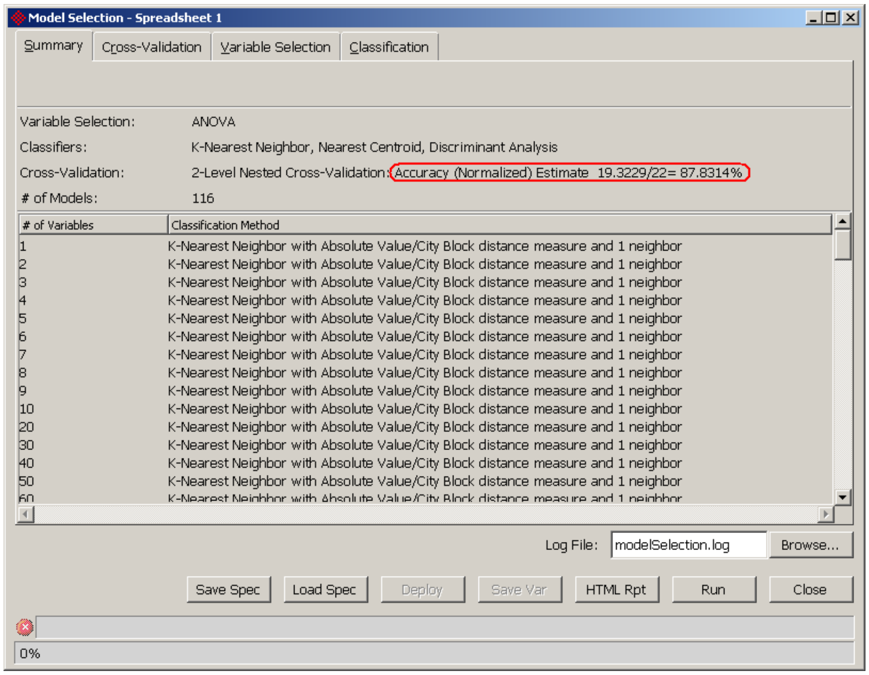
- The data used in the tutorial can be downloaded from http://www.partek.com/~devel/modelSelectionTutorial.zip
- Unzip the tutorial data to a local folder, such as *C:\Partek Tutorial Data\Model Selection*

A classification model has two parts: the variables to use and the classifier to use. The Partek Model Selection tool uses cross-validation to choose the best classification model and give the accuracy estimate of the best model when testing on new data.

1-Level Cross-Validation is used to evaluate multiple models and pick the best model to deploy. 2-Level Cross-Validation is used to give the accuracy estimate. For this exercise, a 2-Level Cross-Validation will be performed.

- Start *Partek*
- Select **File > Open…** from the Partek main menu
- Browse and open **C:\Partek Tutorial Data\Model Selection\trainingSet.fmt**. Note: This data has 11 Down syndrome samples and 11 Normal samples and ~22K genes. Its purpose is to demonstrate the usage on the Partek Model Selection tool. It is not for use in diagnostic procedures. Column *"2. Type"* will be used to train and predict. The Partek Model Selection tool requires this column to be before all the numeric variable columns.
- Select **Tools > Predict > Model Selection** from the Partek main menu
- On the *Model Selection* dialog, select **Load Spec** and choose **C:\Partek Tutorial Data\Model Selection\tutorial.pcms**. This is a saved specification that uses ANOVA as the Variable Selection method; K-Nearest Neighbor, Nearest Centroid, and Discriminant Analysis as the Classification methods. 152 models define this model space; a 2-Level 5x5 Cross-Validation will be performed with this model space
- Select **Run**

- Some Discriminant Analysis models would be over fitting with this specification, so select **Run without those models**. Without those models, the model space will have 116 models
- Wait for the *Model Selection* dialog to finish. It will do 25 inner and 5 outer cross-validations. The output from a 2-Level Cross-Validation is the *Accuracy Estimate*. In this case, it is 86.8314%. It means, if one picks the best model out of this model space and tests against new samples, its expected correct rate will be 86.8314%.



Variable Selection: ANOVA

Classifiers: K-Nearest Neighbor, Nearest Centroid, Discriminant Analysis

Cross-Validation: 2-Level Nested Cross-Validation: Accuracy (Normalized) Estimate 19.3229/22= 87.8314%

# of Models: 116

# of Variables	Classification Method
1	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
2	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
3	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
4	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
5	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
6	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
7	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
8	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
9	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
10	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
20	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
30	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
40	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
50	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor
60	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor

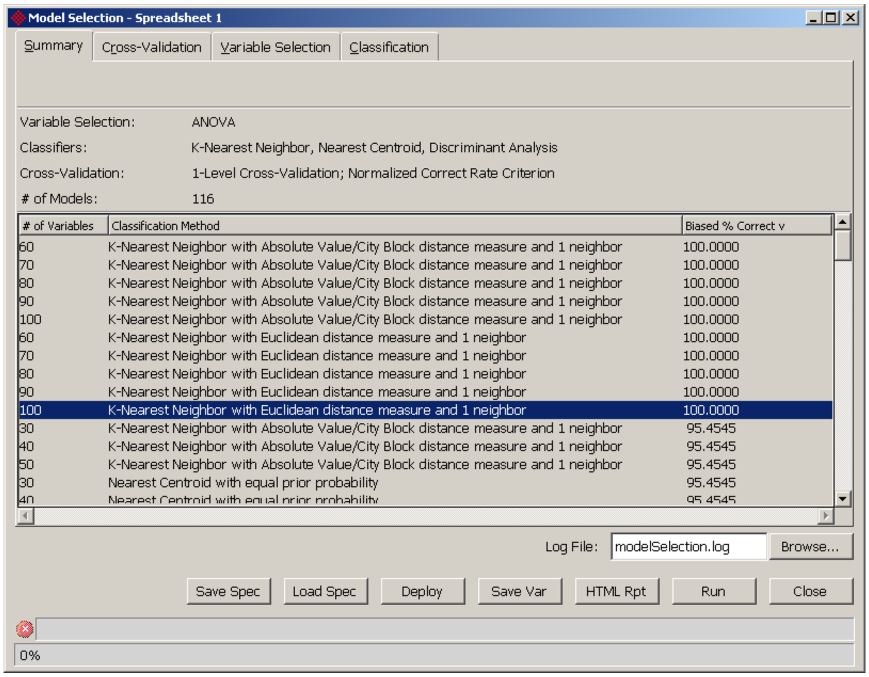
Log File: modelSelection.log

0%

Figure 1: Viewing the Model Selection dialog

Next, perform a 1-Level Cross-Validation to pick the best model to deploy.

- On the *Model Selection* dialog, select the **Cross-Validation** tab and then choose **1-Level Cross-Validation**
- Select **Run**
- Some Discriminant Analysis models would be over fitting with this specification, so select **Run without those models**
- Wait for the *Model Selection* dialog to finish. It will run 116 models with 5-fold cross-validation



Variable Selection: ANOVA

Classifiers: K-Nearest Neighbor, Nearest Centroid, Discriminant Analysis

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

# of Models: 116

# of Variables	Classification Method	Biased % Correct v
60	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	100.0000
70	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	100.0000
80	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	100.0000
90	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	100.0000
100	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	100.0000
60	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	100.0000
70	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	100.0000
80	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	100.0000
90	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	100.0000
100	K-Nearest Neighbor with Euclidean distance measure and 1 neighbor	100.0000
30	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	95.4545
40	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	95.4545
50	K-Nearest Neighbor with Absolute Value/City Block distance measure and 1 neighbor	95.4545
30	Nearest Centroid with equal prior probability	95.4545
40	Nearest Centroid with equal prior probability	95.4545

Log File: modelSelection.log

Figure 2: Viewing the Model Selection, K-Nearest Neighbor with Euclidean Distance

The 116 models produced correct rates ranging from 40% to 100%. Of course, the best model with a 100% correct rate is wanted, but here 100% is a biased correct rate. The expected correct rate, when testing on new samples, will not be 100%, it should be 86.8314% from the 2-level cross-validation. This will be discussed more in the *Common Mistake 2* section below.

When there are tied best models, you may choose to deploy and test all models on the new data and use some kind of voting mechanism when their predictions do not agree. Here, only one model is deployed.

- Select the model with 100 variables, K-Nearest Neighbor with Euclidean distance measure and 1 neighbor, then select Deploy
- In the *Save Variable Selection and Classification Model* dialog, name the file **100var-1NN-Euclidean** and select **Save**

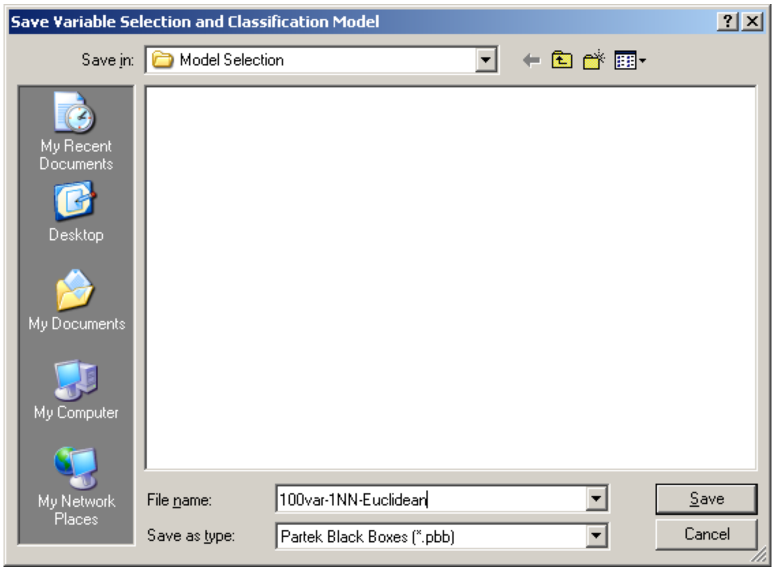
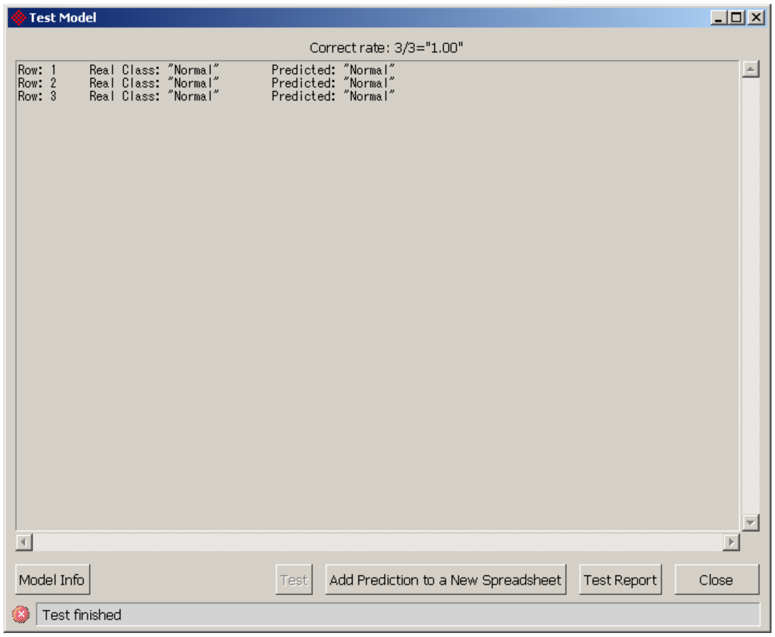


Figure 2: Saving the Variable Selection and Classification Model

Next, this deployed model will be tested on an independent test set.

- Select **File > Open…** from the Partek main menu, browse and open *C:\Partek Tutorial Data\Model Selection\testSet.fmt.* This data has 3 samples, and will be used to demonstrate the usage of the Partek *Model Selection* tool; it is not for use in diagnostic procedures. Column *"2. Type"* will be used for the prediction. The Partek Model Selection tool requires this column to be the same as in the training data and to be placed before all the numeric variable columns, but it is not required that the test data have a real class. For example, the test data may have "unknown" on column 2 *"Type"*
- Select **Tools > Predict > Run Deployed Model** from the Partek main menu. On the *Load Model File* dialog, choose **100var-1NN-Euclidean.pbb** and select **Open**
- On the *Test Model* dialog, select **Model Info**, this will invoke an HTML page. Make sure the test data contains variables listed in the *Variables Used* table and are in the same order
- On the *Test Model* dialog, select **Test**



Correct rate: 3/3="1.00"

```
Row: 1     Real Class: "Normal"        Predicted: "Normal"
Row: 2     Real Class: "Normal"        Predicted: "Normal"
Row: 3     Real Class: "Normal"        Predicted: "Normal"
```

Test finished

Figure 3: Viewing the Test Model dialog

Since column 2 *Type* of the test data has the real class (gold standard), the test result shows the prediction *Correct rate: 3/3 = "1.00"*. It is better than the expected correct rate of 86.8314% received from the 2-Level Cross-Validation; however, if the real class is unknown on column 2 *Type*, the expected correct rate is still 86.8314%.

- Select **Close**

When using the Partek Model Selection tool, there are several requirements:

1. The *Predict On* variable of the training data and the test data should be before all the numeric variable columns
2. The *Predict On* variable of the training data and the test data should be on the same column
3. The test data must have all the variables listed on the *Variable Used* table and must be in the same order as on the table

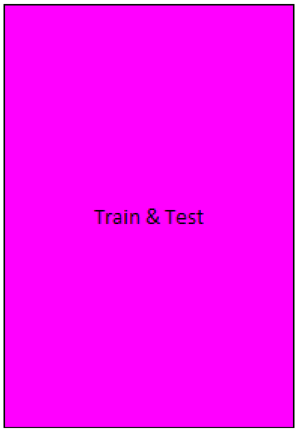
This concludes the common usage section of the Partek Model Selection tool. The rest of the user guide will give more information on cross-validation and give examples of two common mistakes when doing model selection and classification.

Cross-Validation

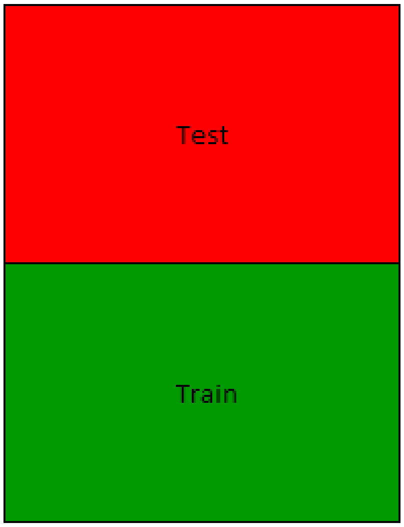
A Classification problem usually has two steps:

- The training step fits a classifier on the training data set with known classes
- The test step applies the trained classifier to predict the classes

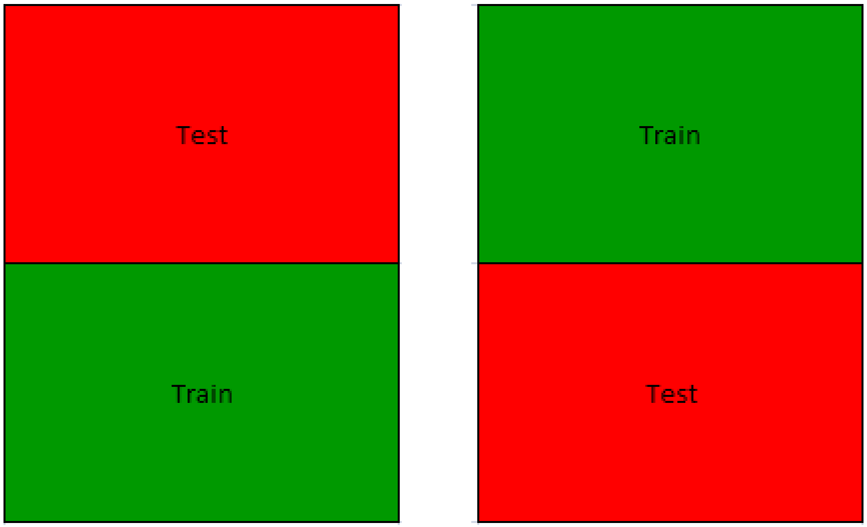
It is crucial that the training set and the test set are independent. Otherwise, one might be ‘over fitting’ the classifier. One example of over fitting is testing on the training set. It is like training students with the real examination questions and testing them on the same questions. The test scores will be biased.



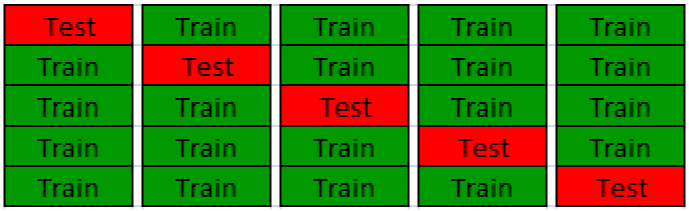
One way of making sure the training set and the test set are independent is to split the whole data into two equal sized partitions. Use one partition as the training set and the other partition as the test set.



But this way is only making use of half of the data. You can actually swap the two partitions to do training and testing again. This is the concept of cross-validation:



You can actually split the data into more partitions. Here's an example of 5-fold cross-validation:



Test	Train	Train	Train	Train
Train	Test	Train	Train	Train
Train	Train	Test	Train	Train
Train	Train	Train	Test	Train
Train	Train	Train	Train	Test

Common Mistake 1: Variable Pre-filtering

The Partek Model Selection tool does the cross-validation data partition first. Variable selection and classification happen inside of cross-validation. One common mistake of doing classification is to do variable selection with all the data, filter the data with selected variables, then split the data into training and test set. This approach is called variable pre-filtering. The test set is being used together with the training set to choose 'good' variables. Those two data sets are no longer independent.

Here, a data set with random numbers will be used to demonstrate the "danger of pre-filtering".

- Start *Partek*
- Select **File > Open…** from the Partek main menu
- Browse and open C:\Partek Tutorial Data\Model Selection\random.txt

The data has 11 Down syndrome samples and 11 Normal samples. There are ~22K genes, but the numbers have been randomized.

- Select **View** > **Scatter Plot** from the Partek main menu. There is no clear separation between Down Syndrome and Normal samples (Figure

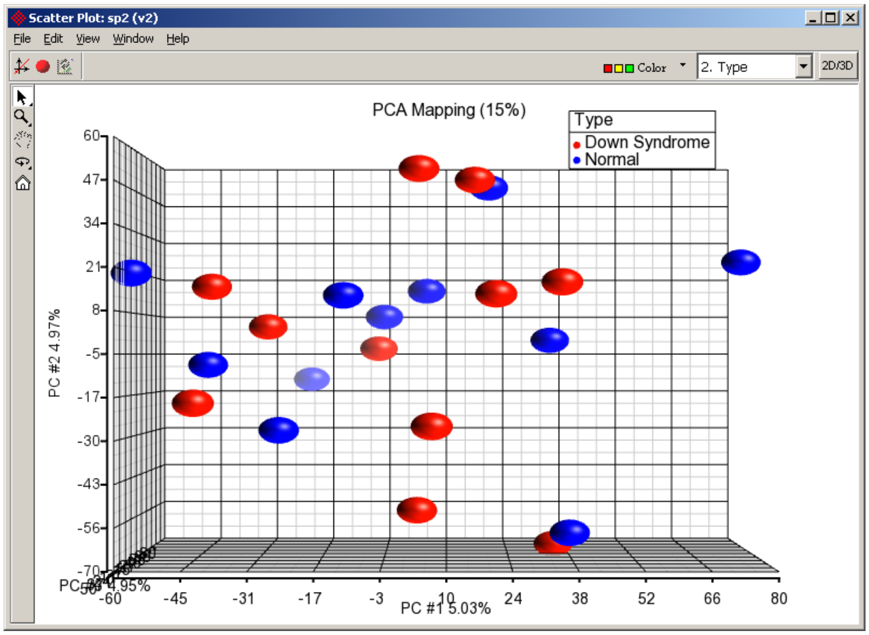
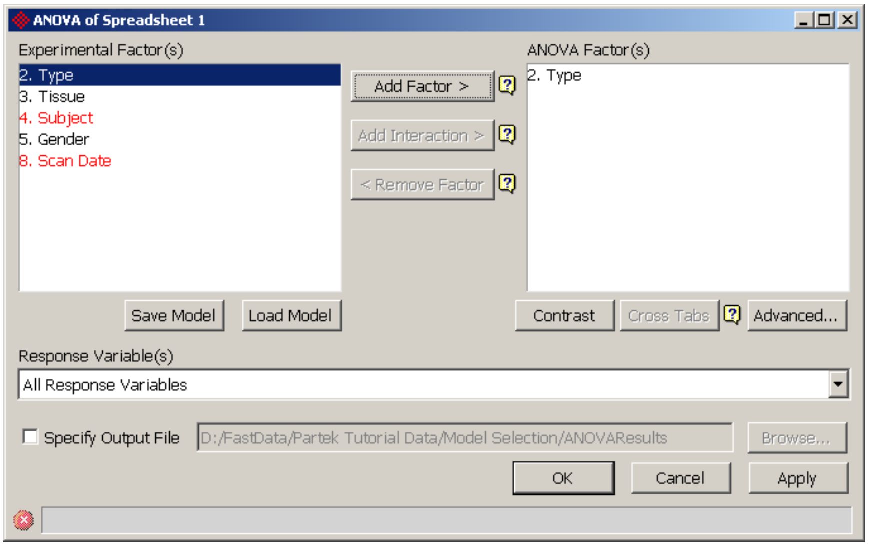


Figure 4: Viewing the Scatter Plot of Down syndrome vs. Normal. Notice there is no clear separation between Type

- Select **Stat** > **ANOVA** from the Partek main menu. On the *ANOVA of Spreadsheet 1* dialog, select **2. Type**
- Select **Add Factor >**, and then **OK** (Figure 5)



Experimental Factor(s)

2. Type
3. Tissue
4. Subject
5. Gender
8. Scan Date

ANOVA Factor(s)

2. Type

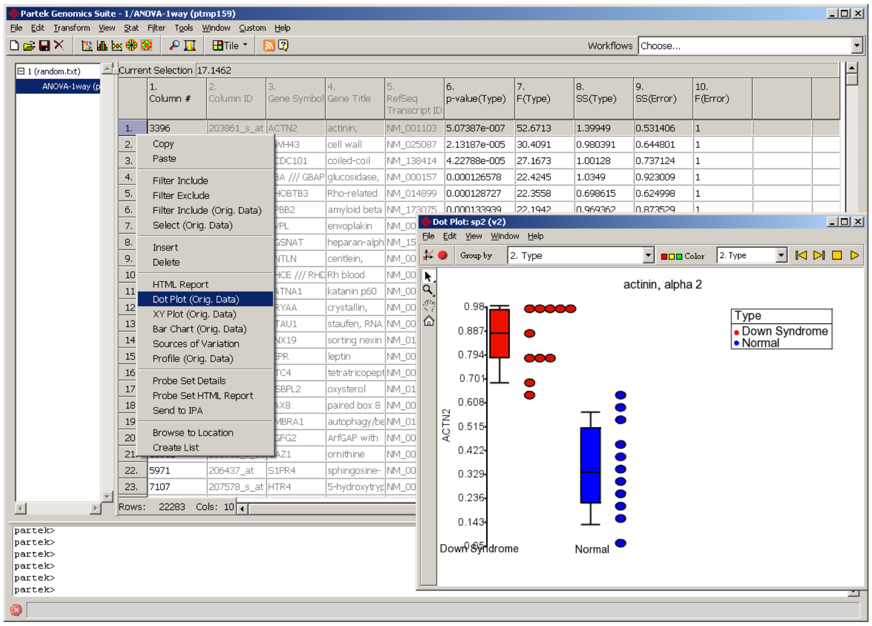
Response Variable(s)

All Response Variables

☐ Specify Output File D:/FastData/Partek Tutorial Data/Model Selection/ANOVAResults

Figure 5: Configuring the Experimental Factors within the ANOVA dialog

- On the resulting spreadsheet, <right-click> on the header of row 1, and select **Dot Plot (Orig. Data)**. The dot plot shows a good separation between the 2 groups. So if the data is filtered to only use this gene, you will probably get a good classifier even on random data. This is called “false discovery”. By chance, you will get some “good” variables with a relatively small number of samples (22 in this case) and a relatively large number of variables (~22K genes), even with random data



Current Selection 17.1462

	1. Column #	2. Column ID	3. Gene Symbol	4. Gene Title	5. RefSeq Transcript ID	6. p-value(Type)	7. F(Type)	8. SS(Type)	9. SS(Error)	10. F(Error)
1.	3396	203861_s_at	ACTN2	actinin,	NM_001103	5.07387e-007	52.6713	1.39949	0.531406	1
2.			WH43	cell wall	NM_025087	2.13187e-005	30.4091	0.980391	0.644801	1
3.			CDC101	coiled-coil	NM_138414	4.22788e-005	27.1673	1.00128	0.737124	1
4.			BA /// GBAP	glucosidase,	NM_000157	0.000126578	22.4245	1.0349	0.923009	1
5.			HOBTB3	Rho-related	NM_014899	0.000128727	22.3558	0.698615	0.624998	1
6.			PBB2	amyloid beta	NM_173075	0.000133939	22.1942	0.969362	0.873529	1
7.			VPL	envoplakin	NM_00					
8.			GSNAT	heparan-alph	NM_15					
9.			NTLN	centlein,	NM_00					
10			HCE /// RHD	Rh blood	NM_00					
11			ATNA1	katanin p60	NM_00					
12			RYAA	crystallin,	NM_00					
13			TAU1	staufen, RNA	NM_00					
14			NX19	sorting nexin	NM_01					
15			EPR	leptin	NM_00					
16			TC4	tetratricopept	NM_00					
17			SBPL2	oxysterol	NM_01					
18			AX8	paired box 8	NM_00					
19			MBRA1	autophagy/be	NM_01					
20			GFG2	ArfGAP with	NM_00					
21.			AZ1	ornithine	NM_00					
22.	5971	206437_at	S1PR4	sphingosine-	NM_00					
23.	7107	207578_s_at	HTR4	5-hydroxytryp	NM_00					

Rows: 22283 Cols: 10

```
partek>
partek>
partek>
partek>
partek>
partek>
```

Figure 6: Invoking the Dot Plot from the resulting ANOVA spreadsheet

For more details about the “danger of pre-filtering”, refer to Ambroise & McLachlan (2002).

Common Mistake 2: Reporting the Correct Rate of the Best Model after a 1-Level Cross-Validation

After running 1-Level Cross-Validation with multiple models, you may want to deploy the model that got the highest correct rate; however, the highest correct rate itself might be biased. The reason is that with multiple (hundreds or even thousands) models running, by chance you may pick a “good” model. Do not expect this model to still achieve that correct rate when testing on new data. Below is an example to demonstrate this concept.

Suppose a class of students is given a quiz with 10 Yes/No questions.

- The 1st student gives 10 Yes’s
- The 2nd student gives 9 Yes’s and 1 No…
- …The last student gives 10 No’s

Since there are 1024 (2^10) answer combinations, if 1024 students systematically enumerate every possibility, one student will answer every question correctly, which will give a biased result, or if instead of systemically enumerating answers, the students randomly pick answers, some students will answer more questions correctly, but the scores will still be biased. With 1-level cross-validation (10-fold, 10 questions), the highest score, 100%, is biased. In the same way, with the 116 models mentioned above, you will get some “good” models by chance.

But if you do a 10x9 2-Level Cross-Validation on all of the students, each pass of the inner Cross-Validation will pick the 2 best students that got 9 questions correct, and test them on the validation question, the 10th question. Of the two best students, one would pick yes, the other would pick no. After averaging them, you would get a 50% accuracy estimate.

So no matter how the students learn/cheat/randomly pick/systemically pick the answers, 2-level Cross-Validation will give you unbiased accuracy estimate.

End of User Guide

This concludes the User Guide. For more information on how to use the Partek Model Selection tool, select **Help > User’s Manual > Chapter 12 Diagnostic & Predictive Modeling** from the Partek main menu.

References

Ambroise, C., and McLachlan, G. “Selection bias in gene extraction on the basis of microarray gene-expression data." PNAS, 2002; 99(10): 6562-6566.