How to use the Machine Learning Plugins for *Mayday*

Stephan Symons

1 Installation

Note: When using *Mayday* via WebStart, you can skip this entire section and proceed to the next section. If you have it installed using an Installer, you can skip the first step, but you must take care of the **external libraries**.
The *installation* of the *Mayday* machine learning plugins is very straightforward. For the installation of *Mayday* itself, please refer to the *Mayday* user guide [1]. To install the machine learning plugins, please copy the plugin JAR, named `mayday-learning-20060430.jar` (the date may change, as new versions emerge) to your *Mayday* plugin folder.
The following external libraries are required for running the plugins:

Library	Purpose	Source
Weka	Drives the Application	`www.cs.waikato.ac.nz/ml/weka/`
Apache POI	Excel Export	`www.apache.org`

Table 1: Required external libraries

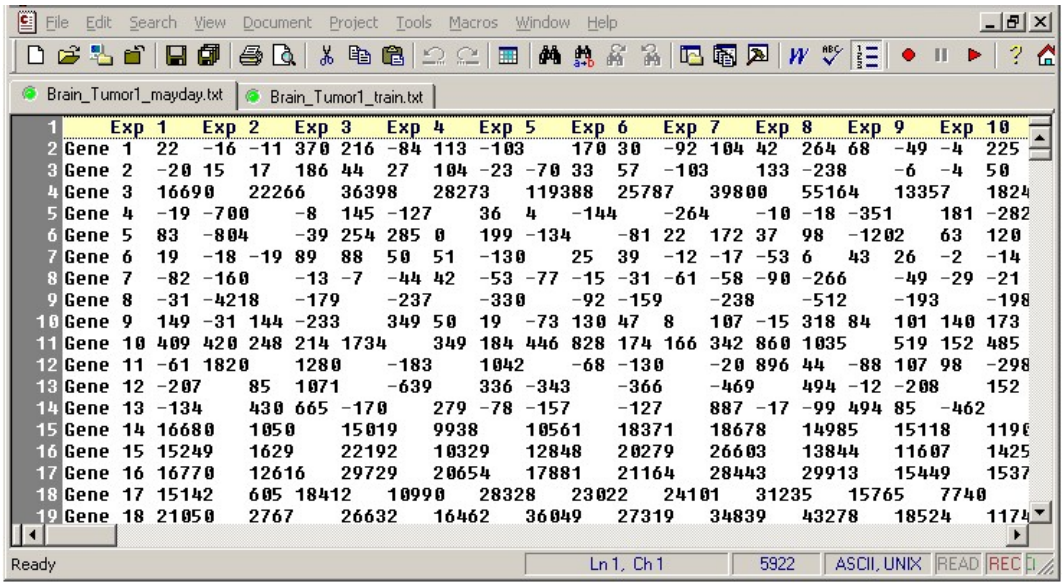
The final step is to set up the external libraries for each plugin. All machine learning plugins described in this work, require the Weka library. The Weka Training and the Weka Classification plugin require also the Apache POI library. To do this, choose the "Plugins..." option from the "File" menu in the *Mayday* main menu. There, locate the plugins in question in the "Data Mining" tab, click on "Preferences" and add the libraries in in the "External JARs" tab. Via the "Add" button, add the path to every required library.

2 Data

Using the *Mayday* machine learning plugins requires a microarray data set that can be read by *Mayday* (delimiter-seperated format, → figure 1) and a file with associations of classes and experiments or genes (in a plain tab-seperated format, → figure 2).

3 Examples

Here, I will demonstrate the use of the *Mayday* machine learning plugins. For a well known microarray dataset, I will describe how to perform feature selection on



	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp 10
Gene 1	22	-16	-11 370	216	-84	113 -103	170	30	-92	104 42
Gene 2	-20	15	17	186	44	27	104	-23	-70	33
Gene 3	16690		22266		36398		28273		119388	
Gene 4	-19	-700		-8	145	-127		36	4	
Gene 5	83	-804		-39	254	285	0		199	-134
Gene 6	19	-18	-19	89	88	50	51		-130	
Gene 7	-82	-160		-13	-7		-44	42		-53
Gene 8	-31	-4218		-179		-237			-330	
Gene 9	149	-31	144	-233		349	50		19	
Gene 10	409	420	248	214	1734		349		184	446
Gene 11	-61	1820		1280		-183			1042	
Gene 12	-207		85	1071		-639			336	-343
Gene 13	-134		430	665	-170		279	-78	-157	
Gene 14	16680		1050		15019		9938		10561	
Gene 15	15249		1629		22192		10329		12848	
Gene 16	16770		12616		29729		20654		17881	
Gene 17	15142		605	18412		10990		28328		23022
Gene 18	21050		2767		26632		16462		36049	

	Exp 6	Exp 7	Exp 8	Exp 9	Exp 10
Gene 1	170 30	-92 104 42	264 68	-49 -4	225
Gene 2	57	-103	133 -238	-6 -4	50
Gene 3	25787	39800	55164	13357	1824
Gene 4	-144	-264	-10 -18 -351	181	-282
Gene 5	-81 22	172 37	98 -1202	63	120
Gene 6	25	39 -12	-17 -53 6	43	26 -2 -14
Gene 7	-15 -31	-61 -58	-90 -266	-49	-29 -21
Gene 8	-92 -159	-238	-512	-193	-198
Gene 9	-73 130	47 8	107 -15	318 84	101 140 173
Gene 10	184 446	828 174	166 342	860 1035	519 152 485
Gene 11	-68 -130	-20	896 44	-88	107 98 -298
Gene 12	-343	-366	-469	494 -12	-208 152
Gene 13	-78 -157	-127	887 -17	-99	494 85 -462
Gene 14	10561	18371	18678	14985	15118
Gene 15	12848	20279	26603	13844	11607
Gene 16	17881	21164	28443	29913	15449
Gene 17	23022	24101	31235	15765	7740
Gene 18	16462	36049	27319	34839	43278

Figure 1: Tab seperated format for microarray data used by *Mayday*.

it, test a set of classifiers on the feature selections using the batch training plugin and build and analyze a reusable classification model.
I will demonstrate the machine learning plugins on the well-known *acute leukemia* dataset published by Golub et al. [2]. The classification task involves discriminating two forms of acute leukemia: acute myeloid leukemia(AML) and acute lymphoblastic leukemia(ALL). This dataset is available for the public and can easily be converted to the *Mayday* format using a spreadsheet application. It consists of a training set and an independent test set. The whole dataset can be considered sufficiently preprocessed for further analysis. The only other preparation is creating the class label files.

4 Feature selection example

The first task is reducing the dimensionality of the dataset. To do so, open the training dataset in *Mayday*, select the "global" probe list and right-click it. Choose from the popup menu in the "Data Mining" → "Classification" submenus the "Weka Feature Selection" plugin. It is has a magnifying glass as an icon.
The "Select class labels" dialog pops up (→ figure 3). Click on the "File" radio button, and use the "Browse" button to locate the class label file in a (operating system dependent) file chooser dialog. It is a good idea to keep the class label information as meta information. Check the "Add as a new MIO group" to have this done. Now click on the "Ok" button to continue.
Now the *Mayday* dataset will be converted to a Weka dataset in order to prepare it for feature selection. When this is done, the "Feature Selection" dialog shows (→ figure 3). In this dialog, the user can access all feature selection tools available.
We will use a filter strategy for feature selection, and try for ease of exposition only

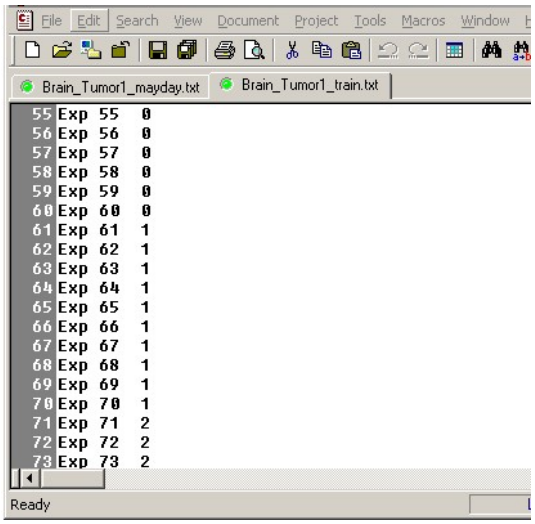


Figure 2: Format for class labels. Each row contains the inforation about one experiment. The first row contains the experiment name, the second one the class label. The experiment names must correspond to the names in the data set. Class labels can be chosed arbitraryly.

the Information Gain evaluator. Select the "Evaluators" radio button, and choose the "InfoGainAttrEval" from the drop down menu. Also, check the "Select best attributes" check box and enter the value "50" in the text field right to it. Remember that the new feature selections are handled as *Mayday* probe lists. You might therefore want to edit the name of the new probe list under "General Settings", for example to "InfoGain 50". You can also set the color to anything other than the default black.
Now, click on the "Ok" button. You will notice, that a new probe list emerged in *Mayday*, with name and color according to your settings and containing the best 50 probes according to Information Gain.
Repeat this procedure with some other probe list sizes. For this purpose, set the "Select best attributes" to other values, for example to 250 and 1000. Update the name and color accordingly. Experiment with other settings. Click on "Ok" to create each feature selection. Click on "Cancel" to close the dialog.
This is a good opportunity to save our work. Use the QuickSave function located in the DataSet menu. It has a red floppy disk as an icon. The whole state of *Mayday*, including all probe lists and MIOs is saved to a predefined file.

5 Batch training example

Now that we have some feature selections, we should see what classifier is the best on this dataset. The mass training plugin is the best for this purpose. Select all probe list you created during feature selection and run the "Weka Batch Training" plugin. You will again be queried for the class labels. This time, choose the MIO group you before feature selection by clicking on the "MIO Group" and choosing the "Class Labels" MIO group. Click on the "Ok" button to continue.

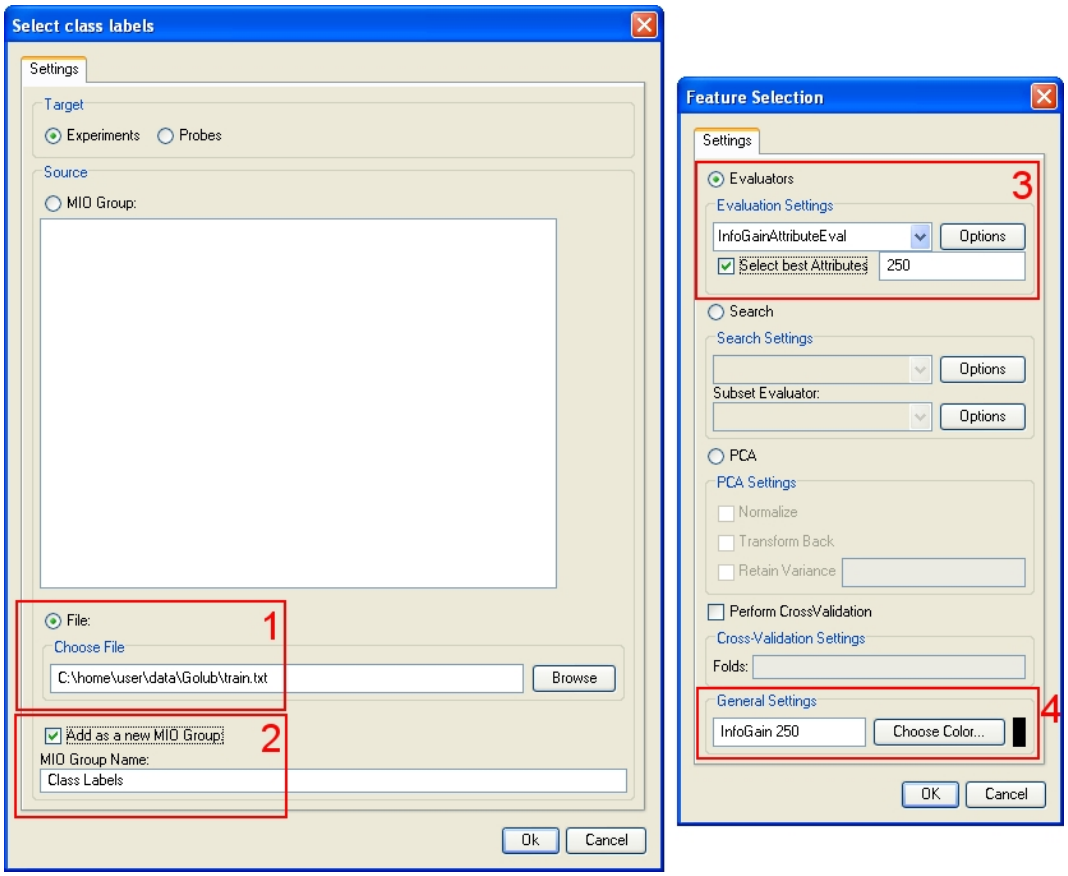


Figure 3: "Class labels" and "Feature Selection" dialog. (1): Select the file; (2): Save the class labels as MIO; (3) Select the InfoGain Evaluator and 250 best attributes; (4): Name the new probe list.

Now, the "Batch Training" dialog shows (→ figure 4). In this dialog, you can create a list of classifiers with different settings, We will test four classifiers in the following: a linear SVM, a kNN classifier with $k = 3$, a C4.5 decision tree and a One Rule. First, we will add the linear SVM. Locate the SVM in the tree structure on the left (try the "Functions" branch). The name and default setting of the classifier is displayed in the "Current classifier field on the upper right. We do not need to change the settings of this classifier, and therefore can just click on the "Add" button. Then, the SVM is added to the list at the bottom of the dialog. Note that classifiers in the list can be edited and removed. The list can be saved to a file for later use.
Next, a kNN classifier (in the "Lazy" branch) should be added. The default parameter for k is 1. Edit this by clicking on the "Options" button and set the "KNN" parameter to 3. Also add this classifier to the list. Next add a C4.5 ("Trees") and a One Rule ("Rules") classifier to the list.
The classifier list is now ready. The default evaluation procedure is ten-fold cross validation. The absolute number of errors will be reported. These settings are reasonable for now.
Click on "Run" to start the batch training. Now, each classifier is trained and evaluated on each selected probe list using 10-fold cross validation.

Eventually, a "Batch training result" dialog (→ figure 4) emerges. The results are presented in the table. It seems, that the SVM is the best classifier in the test field, while the C4.5 is the worst. The minimum errors were made on the "InfoGain 250" probe list.

6 Training and Classification examples

In the previous section, we learned which classifiers should work good on the dataset. Now, we want to produce a classifier which we can use to classify new data. To do this, we need to use the "Weka Training" plugin. Based on the results of the previous chapter, it is best to run it on the "InfoGain 250" probe list. Doing so, the user is queried again for class labels. Again, use the earlier prepared MIO group.

In the "Training" dialog, single classifiers can be trained. In order to demonstrate the diagnostic tools offered by this plugin, it is best to start with a C4.5 decision tree, because the results of the SVM are extremely good, an therefore somewhat boring. Select this classifier from the tree and click on "Start Training". When the training is completed, a trained classifier of type C4.5 with all properties as set before, is listed in the "Previously trained classifiers" list and the "Details..." dialog shows. This dialog provides a variety of valuable information about the classifier.

From this diagnostic plots (→ figure 5) we can see that the C4.5 is pretty good, but the SVM may yet be better. Select the SVM classifier from the tree and train it. The diagnostics shown in the are "Details..." dialog are indeed better for the SVM than for the C4.5.

We now want to use the SVM classifier to classify a new dataset. Select the SVM in the "Previously trained classifiers" and choose "Save as Meta Information". Then, close the dialog.

Open the independent test set in *Mayday*. Remember that the trained classifiers know what genes they are trained on and select their genes automatically when used for classification. Therefore, run the Classification plugin on the "global" probe list. When asked for the class labels, of course none are required (simply click on "Ok"), but can be provided if available as they are useful to compare the results of the classification with the actual classes.

All classifiers saved as MIOs are automatically available in the "Classification" dialog. Additional classifiers can be loaded via the "Load" button or via drag and drop. To classify the dataset with a classifier, select it from the list: the details of the classifier are displayed in the "Classifier" field. Click on the "Classify" button to start the classification. The classification results are displayed in tabs at the bottom of the dialog.

Finally. Save the results as a spreadsheet file by right-clicking on the results table and choose "Export predictions to Excel" button. You now have successfully classified a new dataset and exported the predictions for further use.

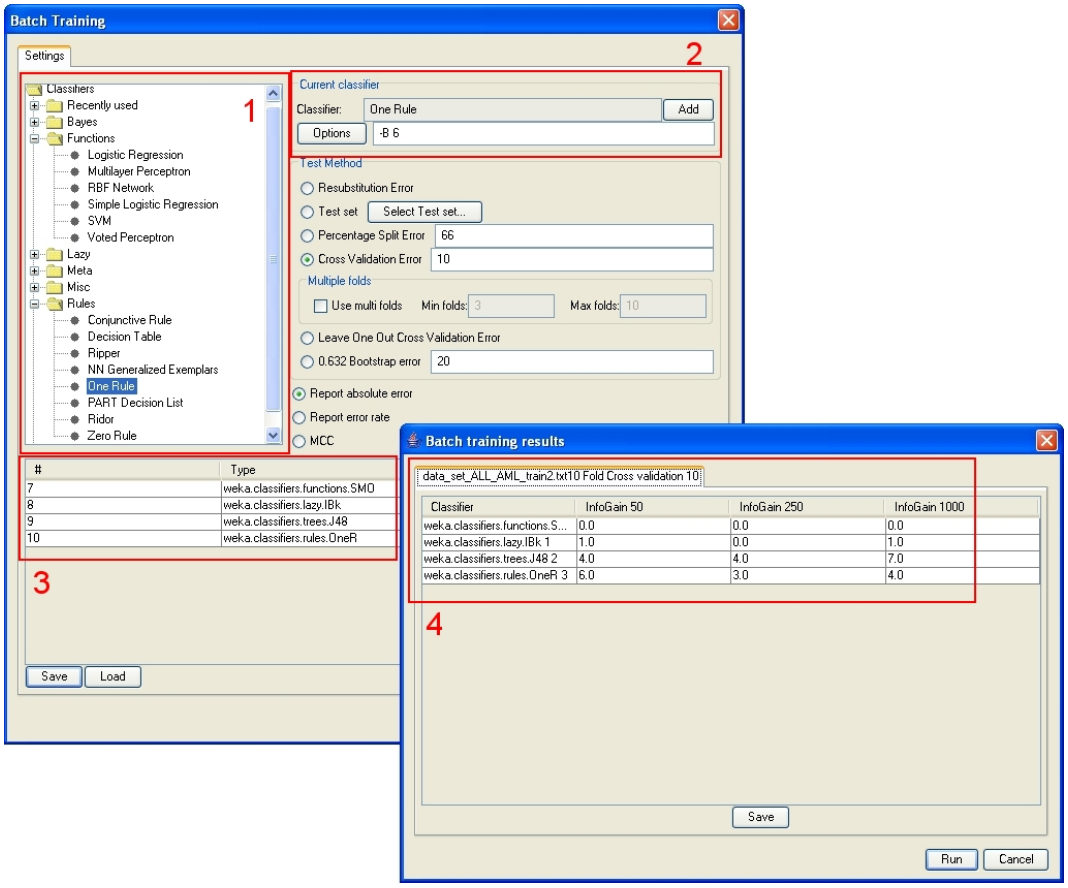


Figure 4: "Batch Training" dialog and "Batch Training Results". (1): Select classifiers from this tree; (2): Edit options and add the classifier to the list; (3): List of classifiers to test (4): Result of the tests. The lists show the Java class names of the classifiers.

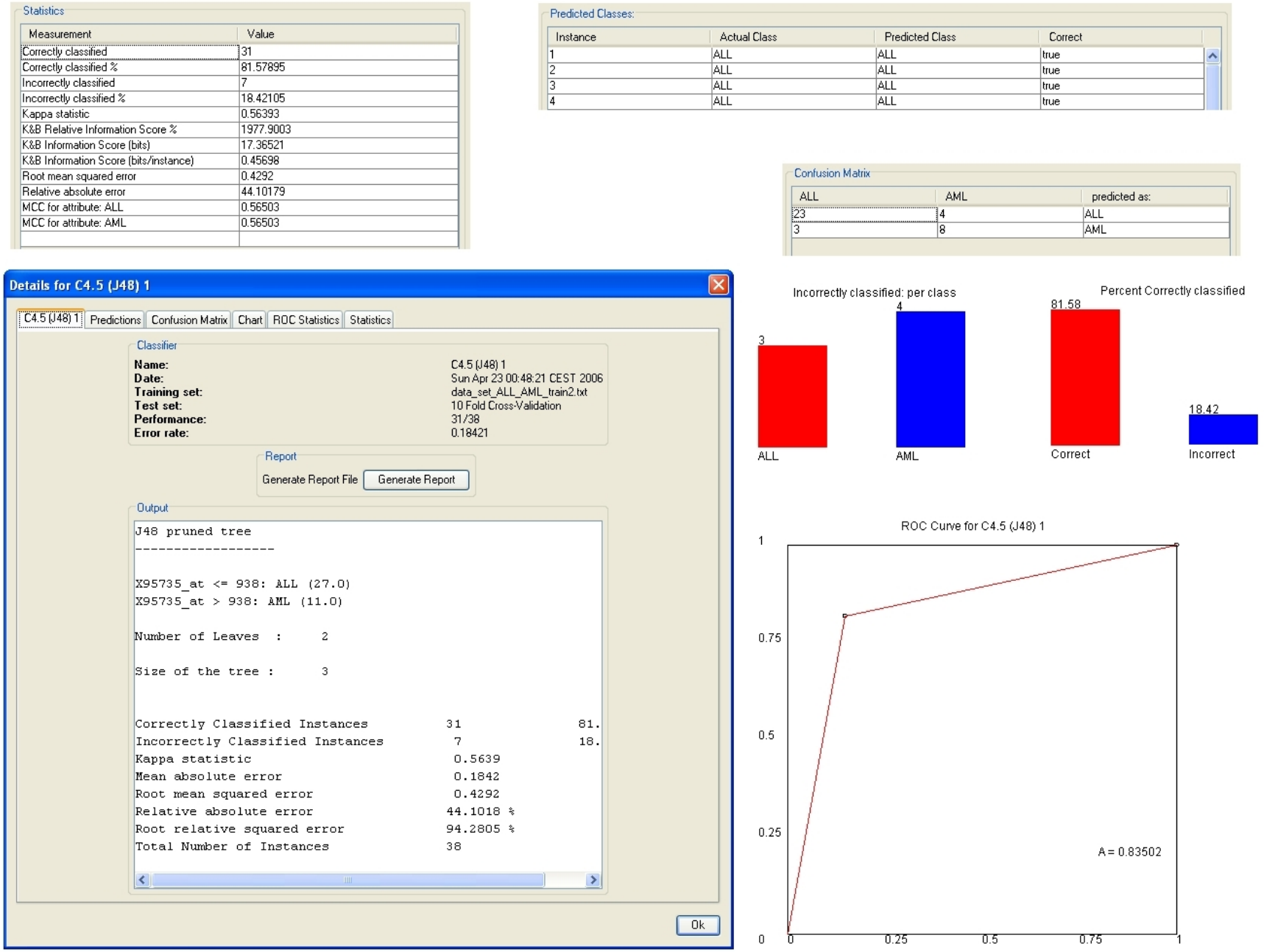


Figure 5: Classifier evaluation of the C4.5 classifier. The figure shows all diagnostic output for the classifier: (in clockwise order) statistics, predictions on the training set, confusion matrix, charts, ROC curve, overview.

References

[1] Janko Dietzsch, Nils Gehlenborg, Stephan Symons, Matthias Zschunke, and Kay Nieselt. *Mayday User Guide*, 11 2005.

[2] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caliguri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

This Mayday How-To was written and edited by Stephan Symons. If you have comments or questions please contact the author via email, *symons@informatik.uni-tuebingen.de*. The latest version of this document can be found at *http://www.zbit.uni-tuebingen.de/pas/mayday*.
