MBF_CLASSIFY
USER MANUAL

Mac Biophotonics Facility

CONTENTS

1. INTRODUCTION

Machine learning is a technique that can provide good classification results for objects seen in microscopic images. There exist many methods by which machine learning can be accomplished and every method makes use of a supervised classifier. A supervised classifier takes a training set consisting of examples of each class and assigns a particular class to an unknown input.

MBF_Classify is based on the same approach and uses three supervised clustering methods namely KNN (K Nearest Neighbors), SVM (Support Vector Machines) and Neural Network to generate classes for every unknown object in the data set. However, MBF_Classify can also be used to collate a training set from a mixed population of images without the user having to manually assign a class to the images.

2. WHAT WILL THE SOFTWARE DO?

2.1 INITIAL CLASSIFICATION RUNS

MBF_Classify allows the user to run a series of classifications over the same data set and save the results for later use. It systematically cycles through different samples of control objects, feature reduction algorithms, number of features kept from the feature reduction, and classifiers. The best performing classification scenario for the given control data set is identified and applied to cluster the unknown data. The underlying algorithm of MBF_classify is based on the idea of systematically testing all the possible classification scenarios applied to the control data and then picking the optimal scenario. Although the

optimal scenario is selected each initial classification run uses only a selection of the control data. Therefore, when the data set is large enough several different initial classifications should be used to identify the most robust feature set.

There are 4 variables in each classification scenario:

1) The size of the control set (values from 25 to a user defined number in steps of 25),

2) Feature reduction method (PCA, KS, SDA),

3) Number of features to keep after feature reduction, and

4) The classification method (KNN, SVM, Neural Network).

The algorithm selects one process from each step (e.g. for the feature reduction step it will choose one of PCA, KS or SDA as a process) to train and test a classifier. The accuracy of the classifier is calculated from the test results and is stored for later use. This process is repeated for the same scenario for a user defined number of replicates. Thus, a series of accuracy values for each possible scenario is recorded. From these accuracy values, an optimal scenario is chosen and then this optimal scenario is used to create a classifier in a final classification stage (see below) and classify the unknown data.

NOTE: KS and SDA are the most frequently used feature reduction methods by MBF_Classify. It has also been observed that supervised clustering is usually performed by using KNN and SVM and very rarely Neural Networks. However, when using 3 controls, KNN is the most preferred method followed by Neural Networks. SVM is rarely used for 3 controls.

2.2 FINAL CLASSIFICATION RUN

Following the analysis of the dataset for a number of times, the user can move ahead to final classification run. During the final run, the software picks only those features that have been used at least 60 % of the times during the initial classification runs of the same dataset and performs a final classification of the dataset. It goes through the same protocol of feature reduction and classification as described for the initial classification result. The results are also saved in the same manner for later use.

3. WHAT IT WILL NOT DO.

The software will not substitute for inaccurate input data. MBF_Classify works only if the data is generated in a certain format from Acapella. The figure below shows the snapshot of a typical data file as per the Acapella script used. The rows in the data file correspond to the objects to be classified. First 15 columns in the data file represent the experimental information and the remaining columns correspond to the features extracted through image analysis.

NOTE: If the software shows an error saying “File not in standard format”, the user can take the following actions:

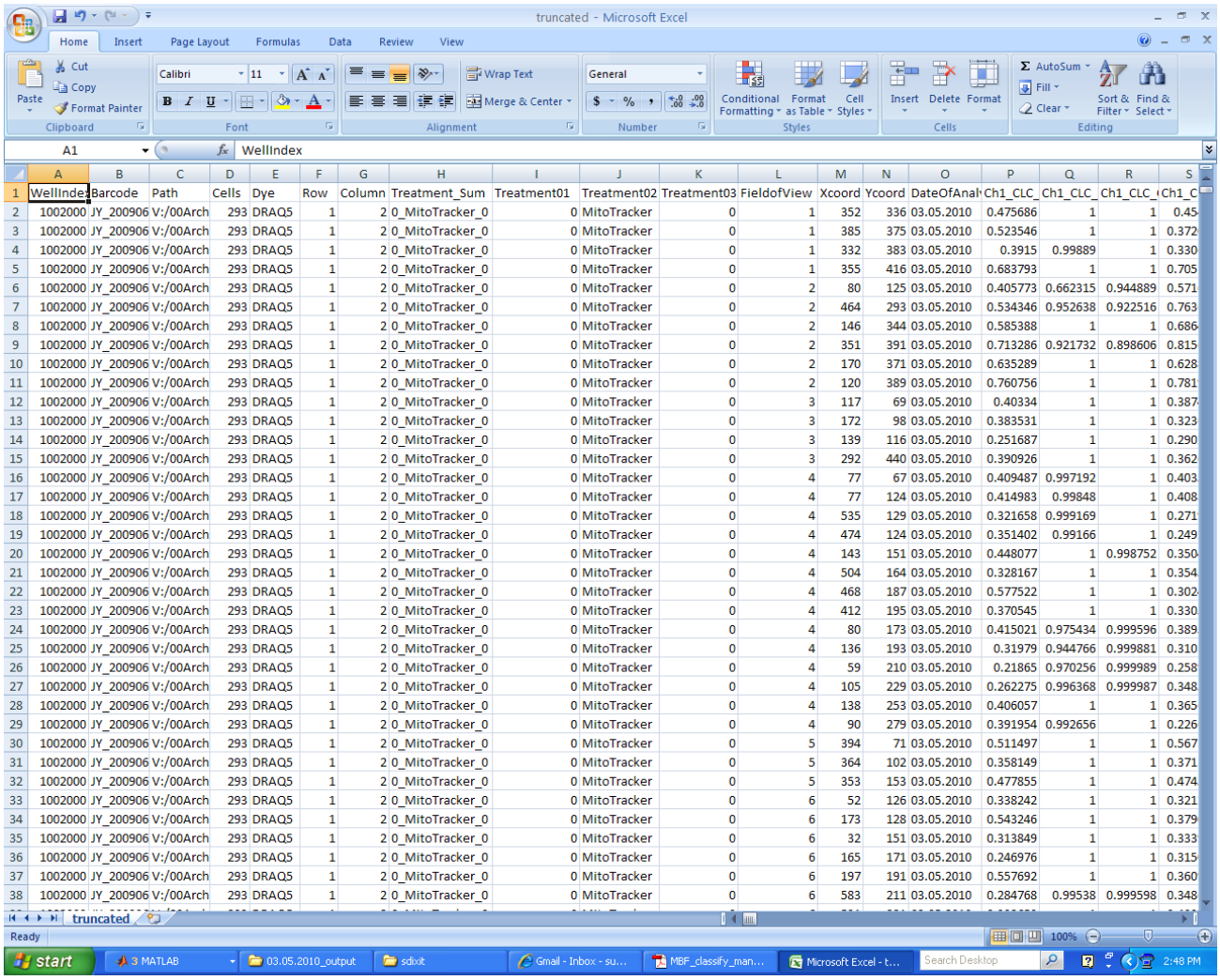
- Check the first 15 columns of the file generated from Acapella. These columns should correspond to the experimental information in the same order as shown in the figure, the eighth column being the treatment sum information.

- Check the naming convention used in the file for all the features starting from column 16 and onwards.
- Check if there are too many clusters of empty rows in the file. However, the software is capable to remove one or two empty rows occurring at some points. This feature has not been tested exhaustively.
- Make sure there are no "Inf" values in the data file. The software can deal with "NaN's" but not "Inf's".

Currently, MBF_Classify allows the user to select between 3 channels (Channel1, Channel2, and Channel3) and 4 feature categories (Morphology, Intensity, Texture, and Colocalization). The nomenclature followed in the data file shown in the following figure is as follows:

- Ch1: Channel 1
- Ch2: Channel 2
- Ch3: Channel 3
- MOR: Morphology
- INT: Intensity
- TXT: Texture
- CLC: Colocalization

NOTE: MATLAB is case sensitive so upper case letters cannot be replaced by lower case letters and vice versa.



WellIndex	Barcode	Path	Cells	Dye	Row	Column	Treatment_Sum	Treatment01	Treatment02	Treatment03	FieldofView	Xcoord	Ycoord	DateOfAnal	Ch1_CLC_	Ch1_CLC_	Ch1_CLC_	Ch1_C
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	1	352	336	03.05.2010	0.475686	1	1	0.45
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	1	385	375	03.05.2010	0.523546	1	1	0.372
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	1	332	383	03.05.2010	0.3915	0.99889	1	0.330
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	1	355	416	03.05.2010	0.683793	1	1	0.705
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	2	80	125	03.05.2010	0.405773	0.662315	0.944889	0.571
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	2	464	293	03.05.2010	0.534346	0.952638	0.922516	0.763
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	2	146	344	03.05.2010	0.585388	1	1	0.686
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	2	351	391	03.05.2010	0.713286	0.921732	0.898606	0.815
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	2	170	371	03.05.2010	0.635289	1	1	0.628
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	2	120	389	03.05.2010	0.760756	1	1	0.781
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	3	117	69	03.05.2010	0.40334	1	1	0.387
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	3	172	98	03.05.2010	0.383531	1	1	0.323
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	3	139	116	03.05.2010	0.251687	1	1	0.290
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	3	292	440	03.05.2010	0.390926	1	1	0.362
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	77	67	03.05.2010	0.409487	0.997192	1	0.403
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	77	124	03.05.2010	0.414983	0.99848	1	0.408
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	535	129	03.05.2010	0.321658	0.999169	1	0.271
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	474	124	03.05.2010	0.351402	0.99166	1	0.249
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	143	151	03.05.2010	0.448077	1	0.998752	0.350
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	504	164	03.05.2010	0.328167	1	1	0.354
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	468	187	03.05.2010	0.577522	1	1	0.302
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	412	195	03.05.2010	0.370545	1	1	0.330
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	80	173	03.05.2010	0.415021	0.975434	0.999596	0.389
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	136	193	03.05.2010	0.31979	0.944766	0.999881	0.310
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	59	210	03.05.2010	0.21865	0.970256	0.999989	0.258
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	105	229	03.05.2010	0.262275	0.996368	0.999987	0.348
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	138	253	03.05.2010	0.406057	1	1	0.365
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	4	90	279	03.05.2010	0.391954	0.992656	1	0.226
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	5	394	71	03.05.2010	0.511497	1	1	0.567
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	5	364	102	03.05.2010	0.358149	1	1	0.371
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	5	353	153	03.05.2010	0.477855	1	1	0.474
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	6	52	126	03.05.2010	0.338242	1	1	0.321
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	6	173	128	03.05.2010	0.543246	1	1	0.379
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	6	32	151	03.05.2010	0.313849	1	1	0.333
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	6	165	171	03.05.2010	0.246976	1	1	0.315
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	6	197	191	03.05.2010	0.557692	1	1	0.360
1002000	JY_200906	V:/00Arch	293	DRAQ5	1	2	0_MitoTracker_0	0	MitoTracker	0	6	583	211	03.05.2010	0.284768	0.99538	0.999598	0.348

NOTE: The order of first 15 columns is important. Note that the data file generated from Acapella should have the same first 15 columns as shown in the figure above. The remaining columns are feature names. Generate the feature names according to the naming convention ***ChannelName_FeatureCategory_....*** In case of the Colocalization feature, the naming convention used is ***ChannelName_FeatureCategory_ChannelName_....***

Examples: Ch1_MOR_Nucleus_area stand for Channel 1 Morphology feature and then the user defined term nucleus area (these terms come from the feature extraction program). That

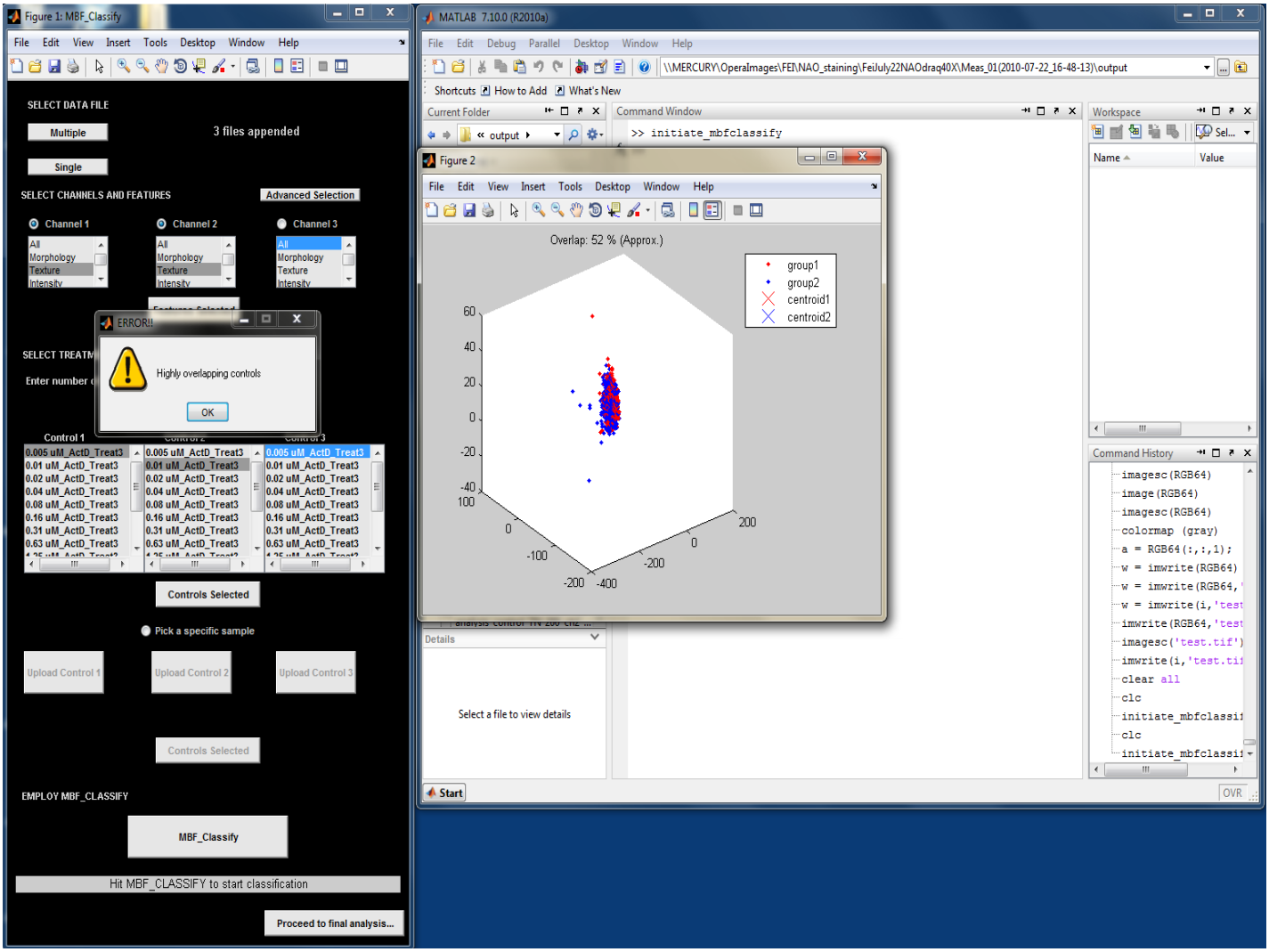
the last part of the name is not fixed permits features to be added or deleted as desired. Ch1_CLC_Ch2_ICQ stands for Channel 1 Colocalization with Channel 2 and in this case indicates that the colocalization feature was an ICQ calculation.

While selecting the channels, the user should take care which channel number corresponds to which color and include only the channels that are required for classification. The user has the option of selecting all three channels or just the two channels required for classification as per the requirements of the experiment. The software also allows the user to select only a single channel.

MBF_Classify has been designed to perform classification using either 2 or 3 controls for training the classifiers. The user must take care that anything other than 2 or 3 controls is not allowed and would generate an error. Also, the software cannot proceed to classification if the controls provided for training have a very high degree of overlap. MBF_Classify provides an approximate demonstration of the amount of overlap in the controls by plotting them in Principle Component Space as shown in the following figure. The approximate percentage of the overlap is also displayed at the top of the PCA plot for the user. A pop up error message also shown, indicates the user to check the controls if it finds them with high amount of overlap. The amount of overlap that is allowed to proceed for classification is anything less than 50%.

NOTE: The software has been designed to take care of high overlaps by stopping them from entering the classification process However, in some cases the data might have an overlap

that is just below 50% but the controls are positioned in such a way that a fair amount of demarcation is not possible. In such a case the software might enter the classification process but report later at the time of classification in the form of an error dialogue box that no feature was found by any of the feature reduction methods to separate the controls. In cases where there is too much overlap because the treated control is heterogeneous (some cells responded and others did not) it may still be possible to classify the images but a KNN single control (described below) may be required.



4. PROTOCOL

The working of MBF_Classify can be divided into two parts. The first is setting up the MBF_Classify inputs in the correct order followed by the second part of employing the classification process. For the convenience of the user, a graphical user interface has been designed that allows the selection of the correct inputs for the classification process.

As mentioned earlier, a two stage classification run is possible with MBF_Classify – Initial classification run and Final classification run. Both the stages have similar steps to follow as mentioned below.

4.1 PRE CLASSIFICATION

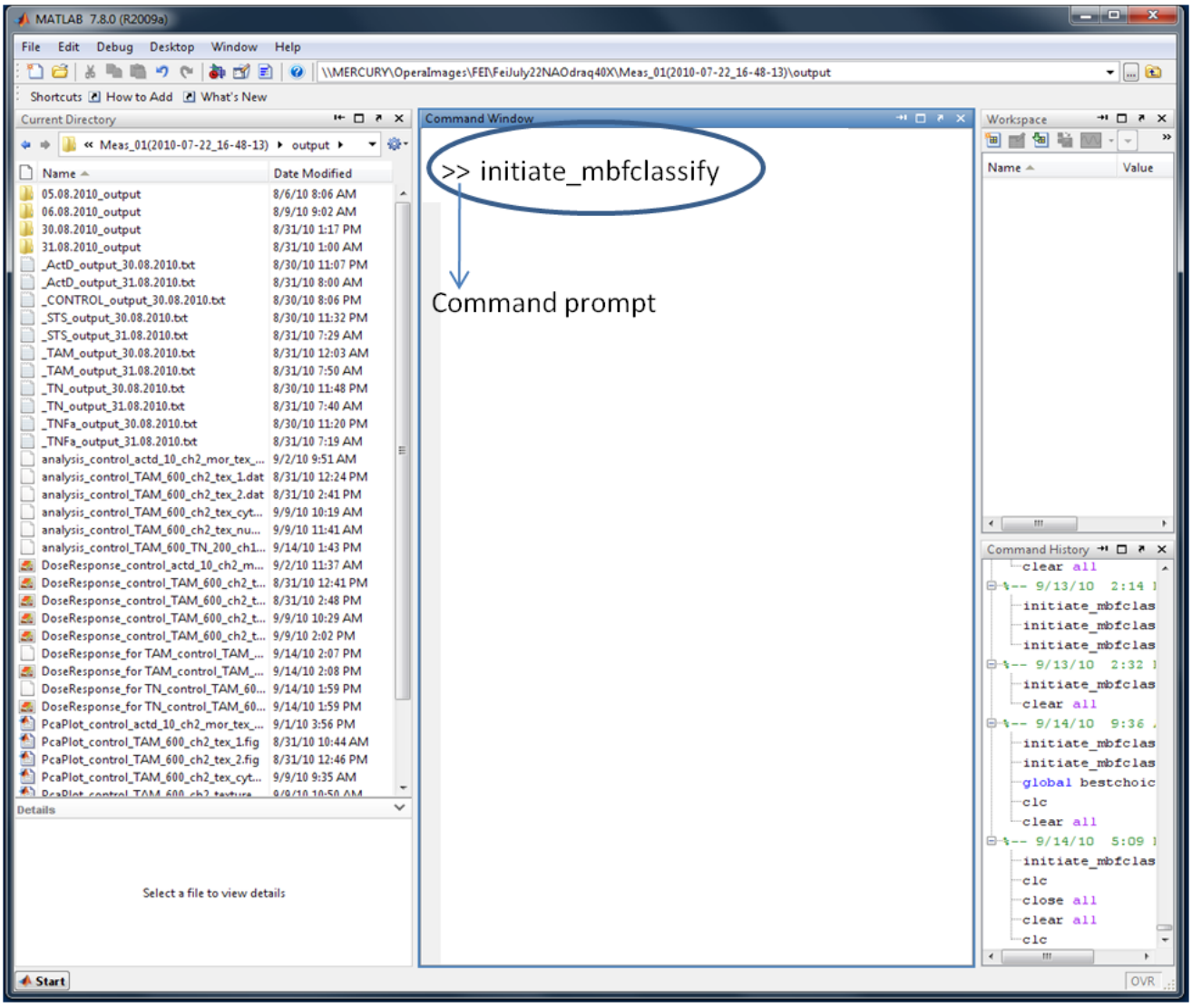
Open MATLAB and set the path of the directory where the program files (.m extension) are saved. The path can be set by using ***"File\set path…"*** and navigating to the folder with the MATLAB program files. This step needs to be followed only when using MBF_Classify on MATLAB for the first time. The path once set is saved in the pathdef.m file of MATLAB for all subsequent runs.

MATLAB REQUIREMENTS: Make sure that the MATLAB version being used has the 3 toolboxes installed before using the software: Neural Network toolbox, Statistics toolbox, and Bioinformatics toolbox. To check the version and toolboxes present in the MATLAB version being used, type ***"ver"*** and press enter on the MATLAB prompt.

4.2 INITIAL CLASSIFICATION RUN

Once the path has been set, the user can start using the software for classification. Follow the steps mentioned below to proceed:

1) Type ***"initiate_mbfclassify"*** at the MATLAB prompt to launch the Graphical User interface for MBF_Classify. The MATLAB prompt and the graphical user interface are shown in the following figure.



2) Press ***"Single"*** on the Graphical User Interface to select a single data file on which the analysis has to be performed, from its specific directory. This file is the output generated from Acapella with ".txt" extension and needs to be in the format specified in section 3. At times, there can be multiple text files generated by Acapella for the same dataset. Hence, to append the files together, the user can hit the ***"Multiple"*** button and select as many files as needed to be appended. Once the data file (files) is (are) selected, the name (number) of the file (files) is (are) displayed on the top right corner of the Graphical User Interface.

NOTE: The size of the data set that can be imported into MATLAB depends on the processor memory. MATLAB can crash and show an error if memory space is low. Generally, a 32 bit processor will not handle data files greater than 2GB in size.

3) The next step is to select the desired features and what channels they correspond to. General procedure is to first select the channel and then its corresponding features. The channel can be selected by clicking on the toggle button followed by feature selection from the respective list. Once feature selection is complete, hit ***"Features Selected"*** to allow MATLAB to process the selected information.

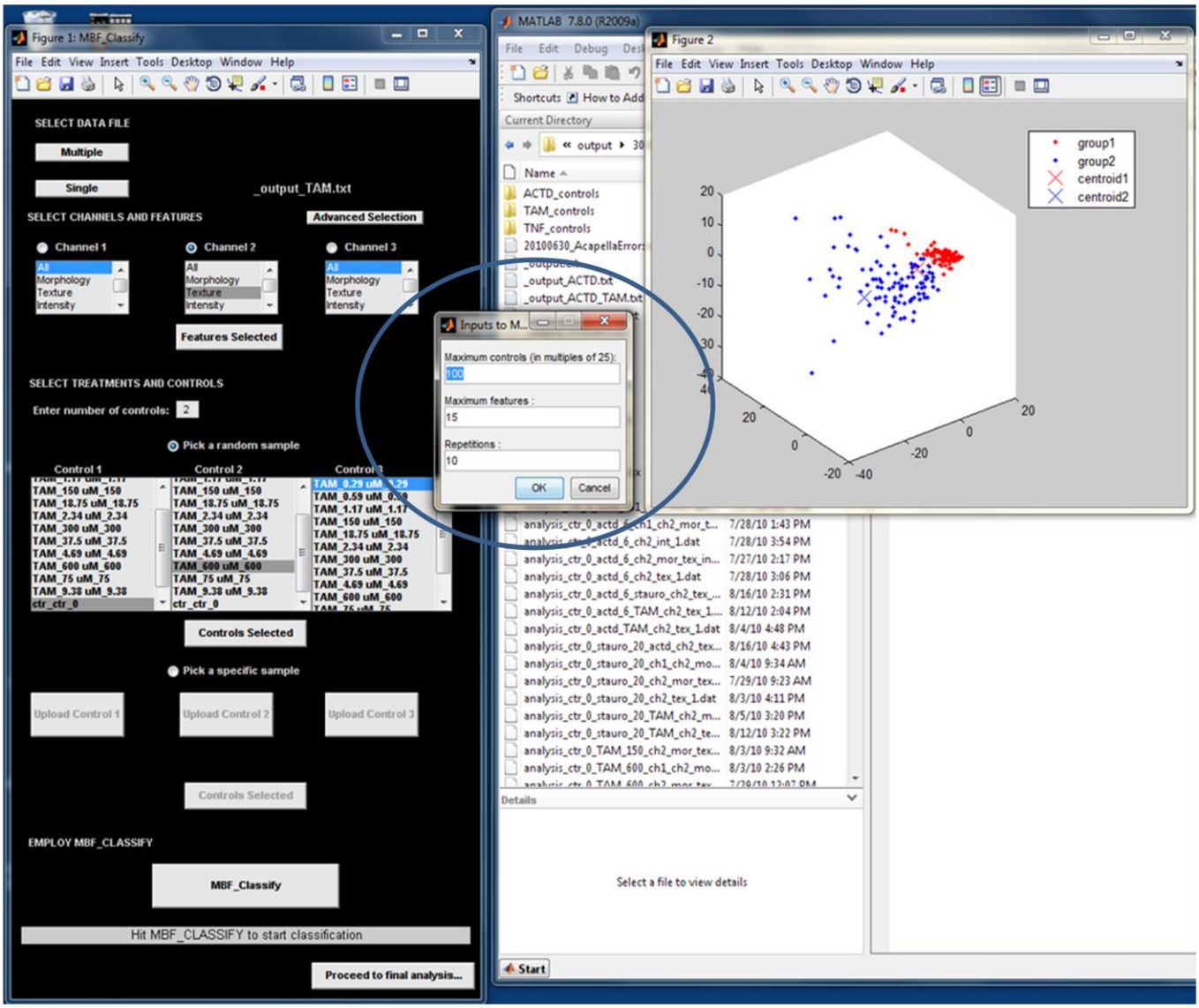
NOTE: To select multiple features press control key and make selection from list.

NOTE: Advanced feature selection tool is also included that allows the user to be even more specific in selecting features. Hence, the user can select features within the major classes of type Texture, Morphology, Intensity or Colocalization as mentioned earlier.

4) Once the processing is complete a list of treatments used in the experiment appears under “Control 1”, “Control 2” and “Control 3”. The user can now specify the number of controls to be used for classification and select the respective controls from the list. Two different treatments under “Control 1” and “Control 2” respectively, should be selected if the user wants to proceed into classification using only 2 controls while for classification with 3 controls, three different treatments under “Control 1”, “Control 2” and “Control 3” respectively, should be selected. Hit ***“Controls selected”*** once the selection of controls is complete. The software also allows the user to upload the specific objects as the controls and proceed towards classification. The control objects to be uploaded should be mat files (extension: .mat). 2 mat files need to be uploaded for running a classification with 2 controls while 3 files need to be uploaded if the user wants to run a three way classification. The specific objects can be selected and saved into mat files using the knn single control algorithm as discussed in Appendix.

   NOTE: Before proceeding towards the selection of controls, make sure the “number of controls” box has been set to the correct number. For example: 2 for selecting two controls and 3 for selecting three controls. For a three control classification, if the user does not change the number of controls to 3 and proceeds towards selecting three treatments per control, the software would completely ignore the third control selected and perform classification using only the first two controls.

5) To start the classification process, hit ***"MBF_Classify"***. As mentioned earlier, MBF_Classify starts checking the controls for the degree of overlap. If the overlap is in permissible limits (below 20 %), it asks the user to input the values for "Maximum Controls", "Maximum Features" and "Repetitions". The window to input values along with the figure for overlap in the controls is shown in the figure below.



NOTE: Controls are selected starting with 25 and stepping up by 25. Default value is set to 100 and generally is a good number for training the classifier. The maximum features to be used must be less than the total features, but in practice, typical values are 15 or less- this allows the computations to be completed in a reasonable length of time. However, the default value has been set to 15. The number of repetitions is exactly the number of times MBF_Classify will cycle through the training and classification process for a given number of controls, feature reduction, number of features and classifier scenario. In practice, 10 repetitions which is the default value, work well without causing the program to require great length of time.

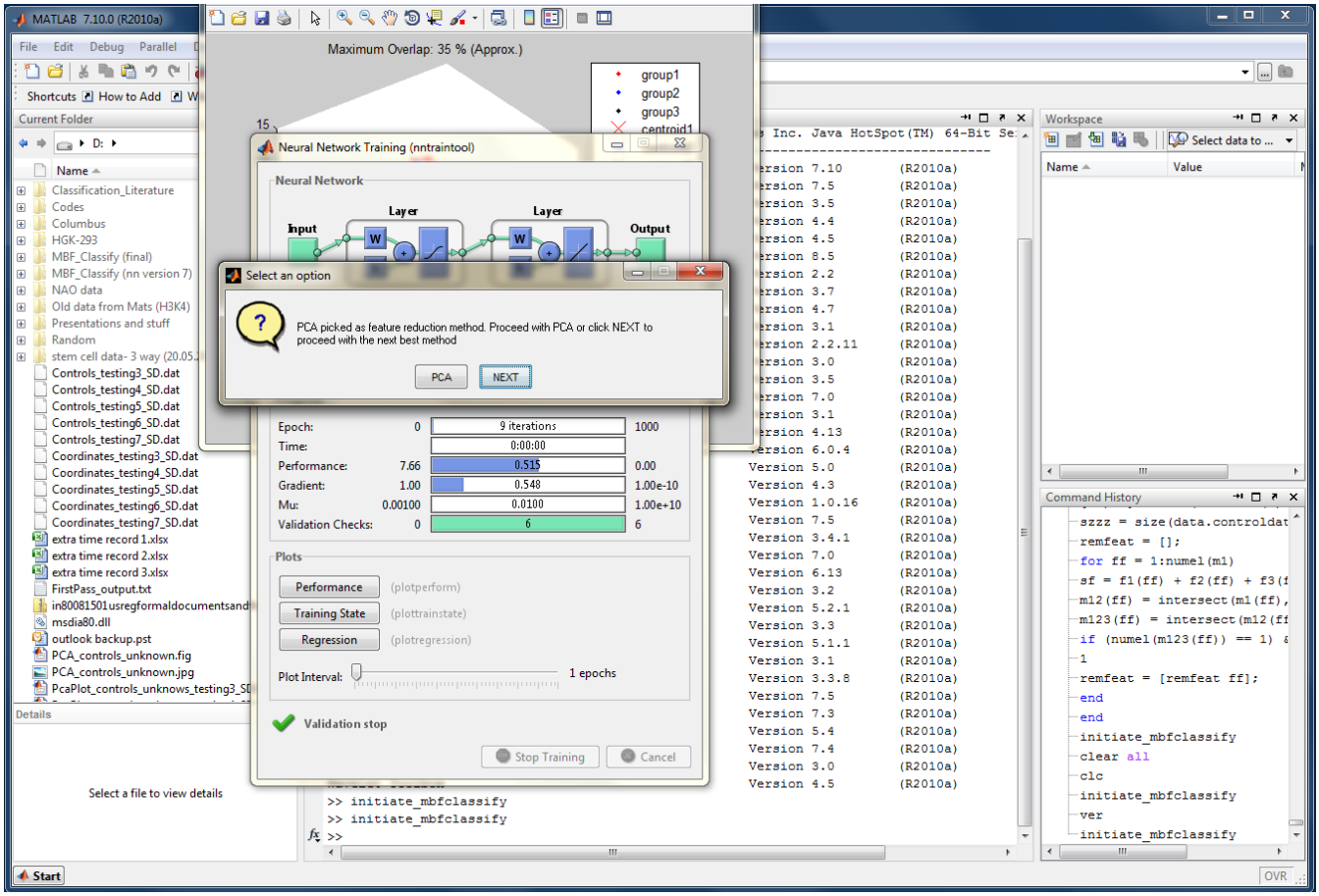
NOTE: The processing times for classification can range from 30 minutes to 6 hours depending on the size of the data set, the total number of features in the set, and the number of repetitions chosen by the user. The processing time can also increase marginally in case of a very high overlap between the controls (approximately between 40 to 50 %)

6) Once the classification is complete, MBF_Classify prompts the user to save the results. The users are encouraged to save the names with experiment number after the underscore to keep track for later use, especially when doing the optional final classification run. The data from these initial classifications can be viewed and used as is. The output is in the same format as the output for the final classification run (see below for how to view and interpret this data). In the initial classification runs PCA can be used as the feature reduction method and to view the data. However, the

output will not include a list of the specific features used as the PCA process combines them linearly. The program permits PCA classification for those users that wish to stop at this stage and not perform a final classification run. Data generated using PCA cannot be used in the final classification because the features are not specified explicitly. However, if MBF_Classify picks up PCA as the best feature reduction method, it would immediately prompt the user as shown in the figure below to chose between carrying on with PCA in which case no feature list would be available or switch to the next best feature reduction method but PCA that was picked after statistical analysis and get the feature list.

NOTE: The instructions to use the Graphical User Interface mentioned above also appear at the bottom of the interface as the user proceeds.

7) **To run another initial classification, close the interface and repeat steps 1 to 5.**



4.3 FINAL CLASSIFICATION RUN

The user should proceed to final classification run only when a minimum of 5 initial classification runs have been completed. Therefore, there should be at least 5 ***"analysis_....txt"*** files each with a different file name before running the final classification. However, 10 initial classification runs are highly recommended. The user can enter the final classification run interface via two routes. First, by clicking on the pushbutton at the bottom right corner of the initial classification run interface named ***"Proceed to final analysis"*** or second, by typing ***"initiate_mbffinalrun"*** on the

MATLAB prompt. Both these steps would open the final classification run interface as shown in the following figure.

The steps needed to be followed are as follows:

1) Press ***"Single"*** or ***"Multiple"*** on the Graphical User Interface to select a single data file or multiple data files, respectively on which the analysis has to be performed, from its specific directory in the manner similar to the one used for the initial classification run. This file is the output generated from Acapella with ***".txt"*** extension and needs to be in the format specified in section 3. Once the data file is selected, the name of the file is displayed on the top right corner of the Graphical User Interface.
2) Press ***"Select analysis files"*** to select the analysis files generated from initial classification run. This button allows the user to select multiple files at the same time by using ctrl or shift keys. Once the set of analysis files have been selected, the corresponding names appear under ***"Analysis file names".*** The interface would automatically update the list of top features that repeated at least 60 % of the times under the title ***"Feature names"***. The user can then select all features in the list or only the top few features to conduct the classification on. Once the feature selection has been made, hit ***"Selected".***

   NOTE: If no feature names appear, please re check the files used for analysis. The possible reasons are that there are no features in common or the initial analysis runs

used PCA as feature reduction method. In either case, try running a few more classification runs to see if any features appear to be commonly used.

NOTE: Multiple feature selection can be done using the ctrl or shift keys.

NOTE: The feature names are arranged in descending order and appear with their respective hit rate as a percentage. A hit rate of 100% means that the particular feature was repeated in all the initial runs and should certainly be used for the final classification run.

3) Once the feature selection is complete, the steps are the same as steps 4 to 6 for the initial classification run explained in Section 4.2.

NOTE: While setting the parameters for final classification run, the user must make sure that the "Maximum Features" input should not exceed the number of features selected under the title ***"Feature names".*** The default value for "Maximum Features" is automatically updated to the number of features selected by the user under the title ***"Feature names"***. It is recommended to perform the classification run using the default values.

NOTE: The software does not allow the user to select three or less than three features for the final classification run.

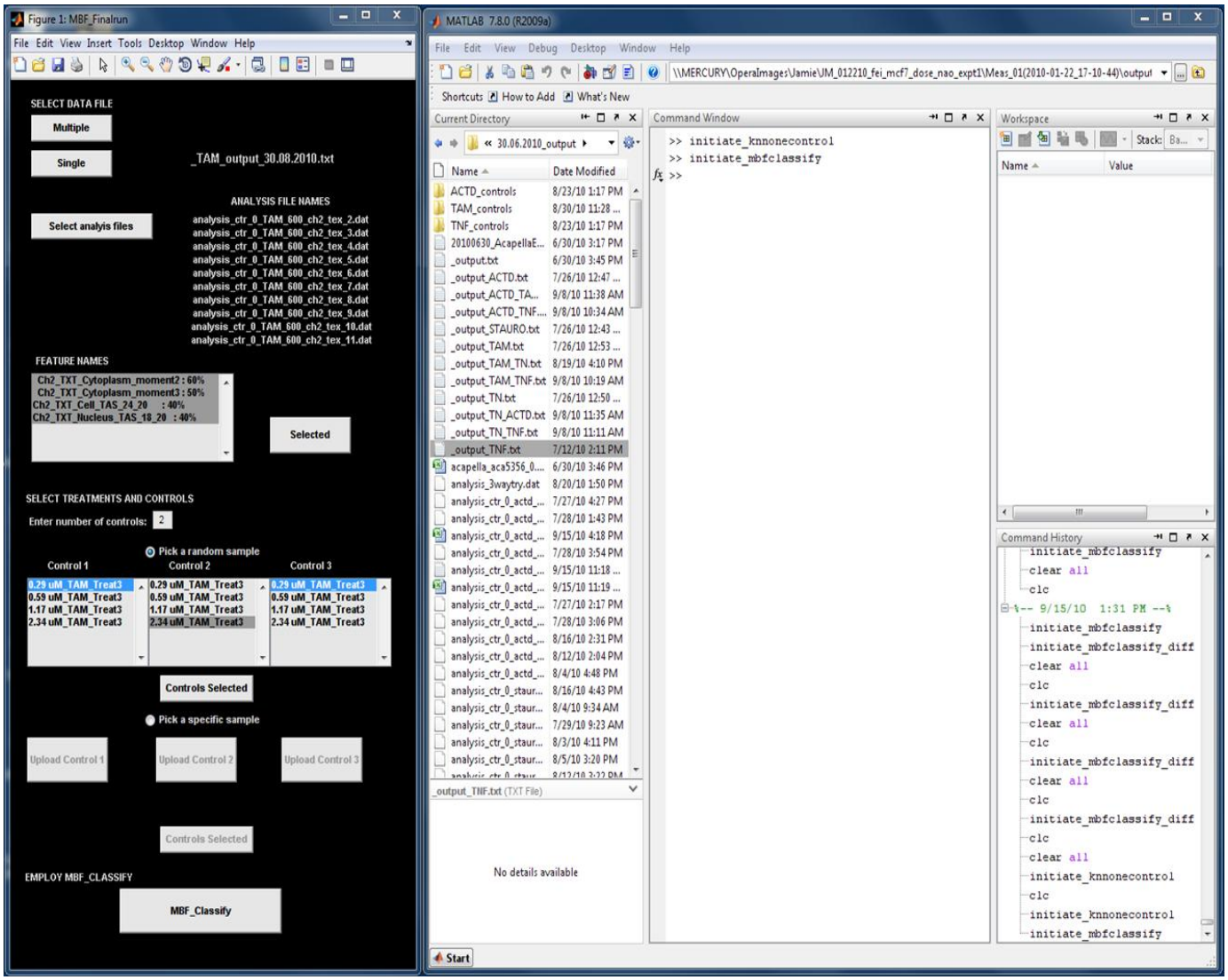


Figure 1: MBF_Finalrun
File Edit View Insert Tools Desktop Window Help
SELECT DATA FILE
Multiple
Single
_TAM_output_30.08.2010.txt
ANALYSIS FILE NAMES
Select analysis files
analysis_ctr_0_TAM_600_ch2_tex_2.dat
analysis_ctr_0_TAM_600_ch2_tex_3.dat
analysis_ctr_0_TAM_600_ch2_tex_4.dat
analysis_ctr_0_TAM_600_ch2_tex_5.dat
analysis_ctr_0_TAM_600_ch2_tex_6.dat
analysis_ctr_0_TAM_600_ch2_tex_7.dat
analysis_ctr_0_TAM_600_ch2_tex_8.dat
analysis_ctr_0_TAM_600_ch2_tex_9.dat
analysis_ctr_0_TAM_600_ch2_tex_10.dat
analysis_ctr_0_TAM_600_ch2_tex_11.dat
FEATURE NAMES
Ch2_TXT_Cytoplasm_moment2 : 60%
Ch2_TXT_Cytoplasm_moment3 : 50%
Ch2_TXT_Cell_TAS_24_20 : 40%
Ch2_TXT_Nucleus_TAS_18_20 : 40%
Selected
SELECT TREATMENTS AND CONTROLS
Enter number of controls: 2
Pick a random sample
Control 1
Control 2
Control 3
0.29 uM_TAM_Treat3
0.59 uM_TAM_Treat3
1.17 uM_TAM_Treat3
2.34 uM_TAM_Treat3
Controls Selected
Pick a specific sample
Upload Control 1
Upload Control 2
Upload Control 3
Controls Selected
EMPLOY MBF_CLASSIFY
MBF_Classify
MATLAB 7.8.0 (R2009a)
File Edit View Debug Desktop Window Help
Shortcuts How to Add What's New
Current Directory
Command Window
>> initiate_knnonecontrol
>> initiate_mbfclassify
>>
Workspace
Name
Value
Command History
initiate_mbfclassify
clear all
clc
%-- 9/15/10 1:31 PM --%
initiate_mbfclassify
initiate_mbfclassify_diff
clear all
clc
initiate_mbfclassify_diff
clear all
clc
initiate_mbfclassify_diff
clear all
clc
initiate_mbfclassify_diff
clc
clear all
initiate_knnonecontrol
clc
initiate_knnonecontrol
initiate_mbfclassify
No details available
Start

5. CLASSIFICATION INTERFACE FOR DIFFERENT CONTROLS AND TEST SETS

There can be cases when the user wants to try a particular set of controls from one data set to classify another set of objects coming from a different data set. Hence, another interface has been designed that allows the user to upload two different text files as the files from where the control set and the test set would be selected, respectively. The protocol to use this interface is similar to the protocol used to run MBF_Classify except for a few changes.

1) Type ***"initiate_mbfclassify_diff"*** at the MATLAB prompt to launch the Graphical User Interface. The MATLAB prompt and the graphical user interface are shown in the following figure.
2) As mentioned for MBF_Classify, the user can upload a single data file or multiple data files by clicking on the ***"single"*** or ***"multiple"*** buttons. However, in this case, the user has to upload two different files or two different sets of files as control and test, separately.
3) The steps ahead of this that is the selction of features and channels, followed by the selection of controls are the same as described for MBF_Classify earlier.

6. INTERPRETATION OF RESULTS

The results of both initial classification run and final classification run consist of two types of files that appear in the "current folder" panel of MATLAB. First is a ***".fig"*** file that contains the PCA plot of the controls used for classification of the data. Second is another ".fig" file that contains the controls and unknowns classified plotted together in the PCA plot. The third file is a ***".dat"*** file that contains the classification results for the test data. The results are saved in two parts (described below). Apart from saving the

results, another ".dat" file is created that saves the information corresponding to the "controls" used for classification. All these ".dat" files can be opened in MATLAB by right clicking on them and selecting the option of "open as text" or as excel file, word file or using WordPad.

The first ".dat" file is labeled as "results_" and includes the following information:

1) FEATURE REDUCTION METHOD: This specifies which method was used for feature reduction before proceeding into classification by MBF_Classify. It can show KS, SDA or PCA as the feature reduction method used.
2) FEATURES USED: The names of the features that were used for classification are specified under this heading.

   NOTE: The feature names are displayed only in the case when KS or SDA have been used as feature reduction methods. However, in case of PCA, no feature names are displayed. This is because PCA uses a combination of various features for classification and not singular discrete features as in the case of SDA and KS feature reduction methods.
3) TREATMENTS: This lists the set of all the treatments used in the data set being analyzed. The controls used for the analysis had been selected from the same list of treatments.
4) SCORES: This gives the scores of the number of objects classified as either of the controls selected earlier. The scores is a matrix with either three or four columns depending on the number of controls used for classification and rows corresponding

to the number of treatments present. In case of two controls, the matrix consists of three columns where the first column gives the total number of objects per treatment, the second column represents the number of objects classified as control 1 per treatment and the third column gives the number of objects classified as control 2 for each treatment. However, in case of three controls, there are four columns in the matrix. The first column being the total number of objects per treatment, second being the number of objects classified as control 1, third being the number of objects classified as control 2 while the last column gives the number of objects classified as control 3 per treatment. Each treatment is presented on a separate row.

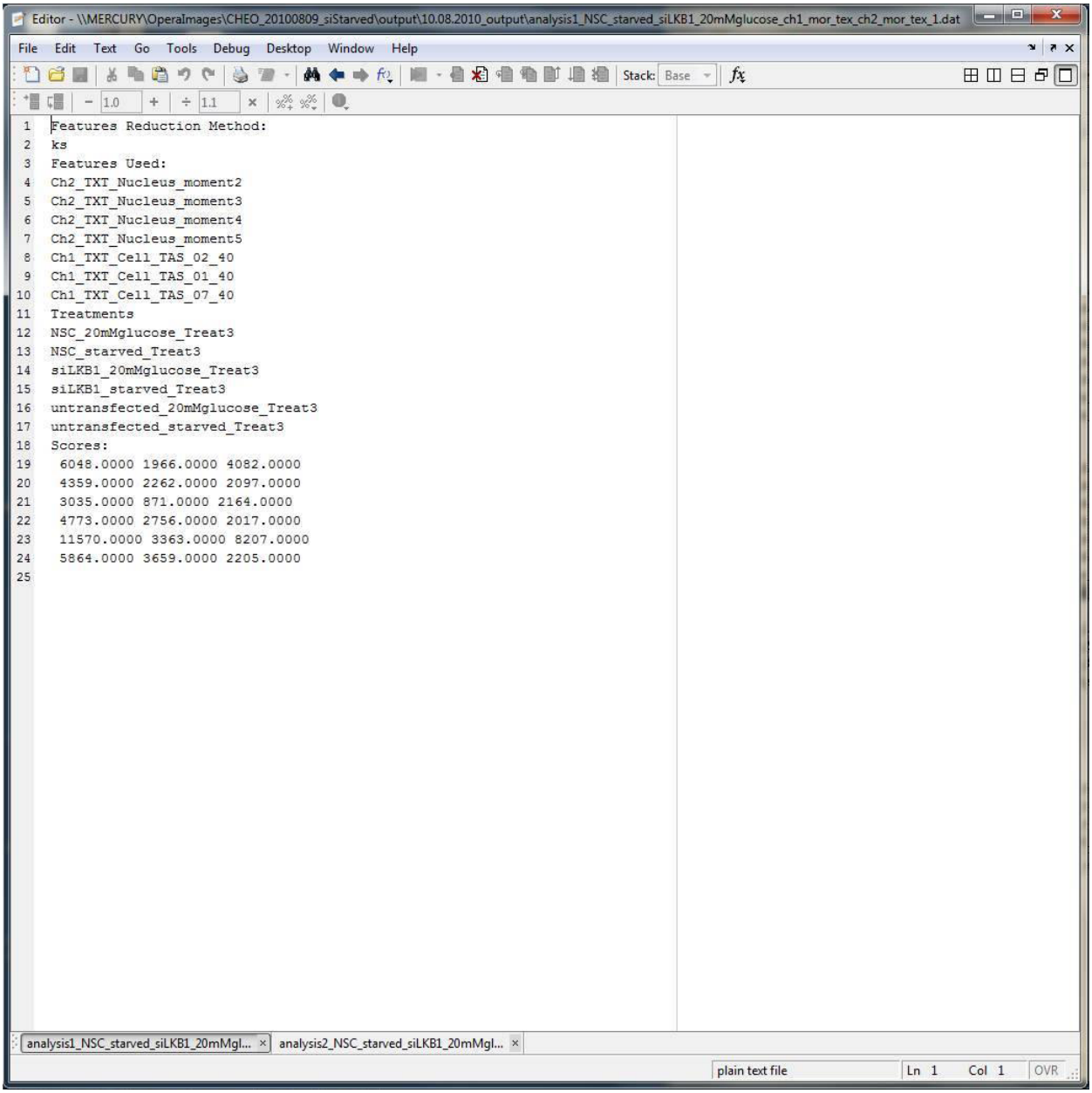
However, the second ".dat" file is labeled as "coordinates_" and includes the information for each cell that was classified as either of the controls selected":

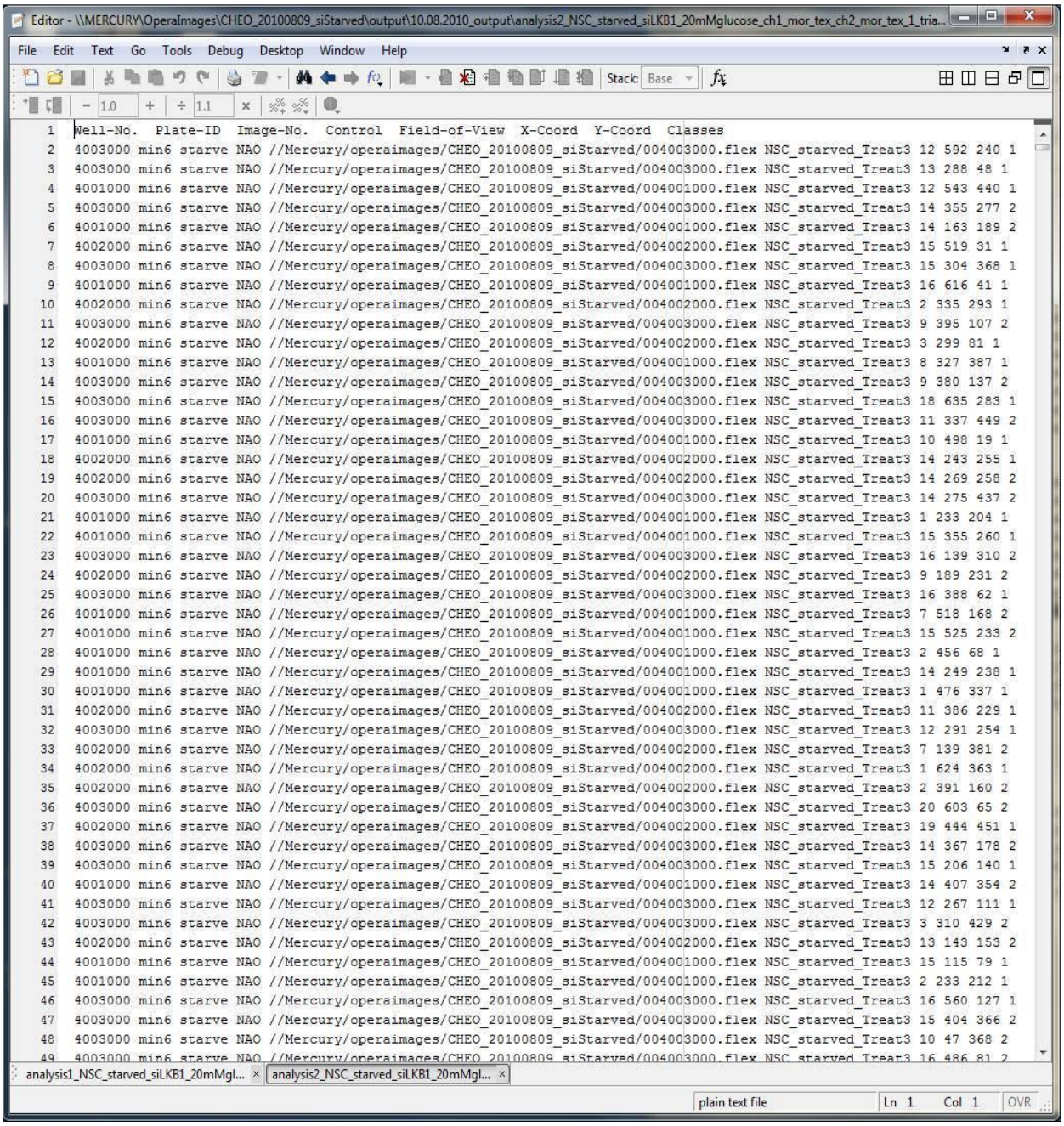
5) The above mentioned variables are followed by a set of variables arranged in a matrix form. The first column in the matrix corresponds to the WELL NUMBER, second column specifies the PLATE ID, third column represents the IMAGE NUMBER, fourth column corresponds to the CONTROL, fifth column representing the FIELD OF VIEW, sixth and seventh columns are for X- COORDINATES and Y-COORDINATES of the object being classified and the last or eighth row specifies the classification result of the cell.

The third ".dat" file consists of the same information as stored in "coordinates_" except that the file is named "controls_" and shows the information corresponding to the controls used for classification.

The following figure shows the three ".dat" files created after the initial classification run. Similar files are created after the final classification run.

NOTE: For files named "coordinates_" and "controls_" the last column that is the class category, 1 represents the type of object selected as control 1, 2 represents the type of object selected as control 2 and 3 if present, represents the type of object selected as control 3.


Editor - \\MERCURY\OperaImages\CHEO_20100809_siStarved\output\10.08.2010_output\analysis1_NSC_starved_siLKB1_20mMglucose_ch1_mor_tex_ch2_mor_tex_1.dat
File Edit Text Go Tools Debug Desktop Window Help
Stack: Base
Features Reduction Method:
ks
Features Used:
Ch2_TXT_Nucleus_moment2
Ch2_TXT_Nucleus_moment3
Ch2_TXT_Nucleus_moment4
Ch2_TXT_Nucleus_moment5
Ch1_TXT_Cell_TAS_02_40
Ch1_TXT_Cell_TAS_01_40
Ch1_TXT_Cell_TAS_07_40
Treatments
NSC_20mMglucose_Treat3
NSC_starved_Treat3
siLKB1_20mMglucose_Treat3
siLKB1_starved_Treat3
untransfected_20mMglucose_Treat3
untransfected_starved_Treat3
Scores:
6048.0000 1966.0000 4082.0000
4359.0000 2262.0000 2097.0000
3035.0000 871.0000 2164.0000
4773.0000 2756.0000 2017.0000
11570.0000 3363.0000 8207.0000
5864.0000 3659.0000 2205.0000
analysis1_NSC_starved_siLKB1_20mMgl...
analysis2_NSC_starved_siLKB1_20mMgl...
plain text file
Ln 1 Col 1 OVR


Editor - \\MERCURY\OperaImages\CHEO_20100809_siStarved\output\10.08.2010_output\analysis2_NSC_starved_siLKB1_20mMglucose_ch1_mor_tex_ch2_mor_tex_1_tria...
File Edit Text Go Tools Debug Desktop Window Help
Stack: Base
Well-No. Plate-ID Image-No. Control Field-of-View X-Coord Y-Coord Classes
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 12 592 240 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 13 288 48 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 12 543 440 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 14 355 277 2
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 14 163 189 2
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 15 519 31 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 15 304 368 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 16 616 41 1
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 2 335 293 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 9 395 107 2
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 3 299 81 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 8 327 387 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 9 380 137 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 18 635 283 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 11 337 449 2
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 10 498 19 1
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 14 243 255 1
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 14 269 258 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 14 275 437 2
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 1 233 204 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 15 355 260 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 16 139 310 2
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 9 189 231 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 16 388 62 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 7 518 168 2
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 15 525 233 2
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 2 456 68 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 14 249 238 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 1 476 337 1
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 11 386 229 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 12 291 254 1
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 7 139 381 2
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 1 624 363 1
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 2 391 160 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 20 603 65 2
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 19 444 451 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 14 367 178 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 15 206 140 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 14 407 354 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 12 267 111 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 3 310 429 2
4002000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004002000.flex NSC_starved_Treat3 13 143 153 2
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 15 115 79 1
4001000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004001000.flex NSC_starved_Treat3 2 233 212 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 16 560 127 1
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 15 404 366 2
4003000 min6 starve NAO //Mercury/operaimages/CHEO_20100809_siStarved/004003000.flex NSC_starved_Treat3 10 47 368 2
analysis1_NSC_starved_siLKB1_20mMgl...
analysis2_NSC_starved_siLKB1_20mMgl...
plain text file
Ln 1 Col 1
OVR

Editor - D:\Controls_testing8_SD.dat

File Edit Text Go Tools Debug Desktop Window Help

	Well-No.	Plate-ID	Image-No.	Control	Field-of-View	X-Coord	Y-Coord	Classes
2	1017000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001017000.tif	K9_GMB3_Treat03	1	596	133	1
3	1004000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001004000.tif	K9_GMB3_Treat03	1	654	965	1
4	1011000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001011000.tif	K9_GMB3_Treat03	1	993	178	1
5	1008000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001008000.tif	K9_GMB3_Treat03	1	1027	515	1
6	1036000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001036000.tif	K9_GMB3_Treat03	1	897	367	1
7	1019000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001019000.tif	K9_GMB3_Treat03	1	908	276	1
8	1015000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001015000.tif	K9_GMB3_Treat03	1	767	926	1
9	1035000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001035000.tif	K9_GMB3_Treat03	1	1219	241	1
10	1010000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001010000.tif	K9_GMB3_Treat03	1	715	768	1
11	1016000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001016000.tif	K9_GMB3_Treat03	1	674	595	1
12	1029000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001029000.tif	K9_GMB3_Treat03	1	549	901	1
13	1004000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001004000.tif	K9_GMB3_Treat03	1	1026	870	1
14	1019000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001019000.tif	K9_GMB3_Treat03	1	616	808	1
15	1002000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001002000.tif	K9_GMB3_Treat03	1	172	179	1
16	1009000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001009000.tif	K9_GMB3_Treat03	1	372	436	1
17	1022000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001022000.tif	K9_GMB3_Treat03	1	209	258	1
18	1019000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001019000.tif	K9_GMB3_Treat03	1	244	629	1
19	1018000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001018000.tif	K9_GMB3_Treat03	1	1214	567	1
20	1017000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001017000.tif	K9_GMB3_Treat03	1	188	249	1
21	1002000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001002000.tif	K9_GMB3_Treat03	1	544	396	1
22	1009000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001009000.tif	K9_GMB3_Treat03	1	750	504	1
23	1011000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001011000.tif	K9_GMB3_Treat03	1	811	180	1
24	1013000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001013000.tif	K9_GMB3_Treat03	1	842	673	1
25	1006000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001006000.tif	K9_GMB3_Treat03	1	309	761	1
26	1005000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001005000.tif	K9_GMB3_Treat03	1	275	905	1
27	1005000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001005000.tif	K9_GMB3_Treat03	1	208	243	1
28	1023000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001023000.tif	K9_GMB3_Treat03	1	903	435	1
29	1010000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001010000.tif	K9_GMB3_Treat03	1	670	347	1
30	1041000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001041000.tif	K9_GMB3_Treat03	1	791	909	1
31	1019000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001019000.tif	K9_GMB3_Treat03	1	249	697	1
32	1007000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001007000.tif	K9_GMB3_Treat03	1	438	451	1
33	1006000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001006000.tif	K9_GMB3_Treat03	1	473	739	1
34	1022000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001022000.tif	K9_GMB3_Treat03	1	912	269	1
35	1004000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001004000.tif	K9_GMB3_Treat03	1	163	379	1
36	1025000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001025000.tif	K9_GMB3_Treat03	1	660	866	1
37	1007000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001007000.tif	K9_GMB3_Treat03	1	786	427	1
38	1010000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001010000.tif	K9_GMB3_Treat03	1	975	306	1
39	1016000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001016000.tif	K9_GMB3_Treat03	1	625	280	1
40	1009000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001009000.tif	K9_GMB3_Treat03	1	821	598	1
41	1038000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001038000.tif	K9_GMB3_Treat03	1	1175	933	1
42	1038000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001038000.tif	K9_GMB3_Treat03	1	617	475	1
43	1022000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001022000.tif	K9_GMB3_Treat03	1	411	647	1
44	1029000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001029000.tif	K9_GMB3_Treat03	1	393	502	1
45	1019000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001019000.tif	K9_GMB3_Treat03	1	1068	129	1
46	1008000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001008000.tif	K9_GMB3_Treat03	1	653	500	1
47	1002000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001002000.tif	K9_GMB3_Treat03	1	300	815	1
48	1009000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001009000.tif	K9_GMB3_Treat03	1	1076	656	1
49	1017000	NoBarcode	V:/MBF_AT_hsc/AT_20110120_GBM/001017000.tif	K9_GMB3_Treat03	1	1132	643	1

MBF_classify.m | final_run.m | controls_unknowns.m | abc.m | initiate_mbffinalrun.m | subsasgn.m | newff.m | MB

plain text file Ln 1 Col 1 OVR

NOTE: This file can be used by Acapella directly to look at the images of the classified cells.

APPENDIX (A)

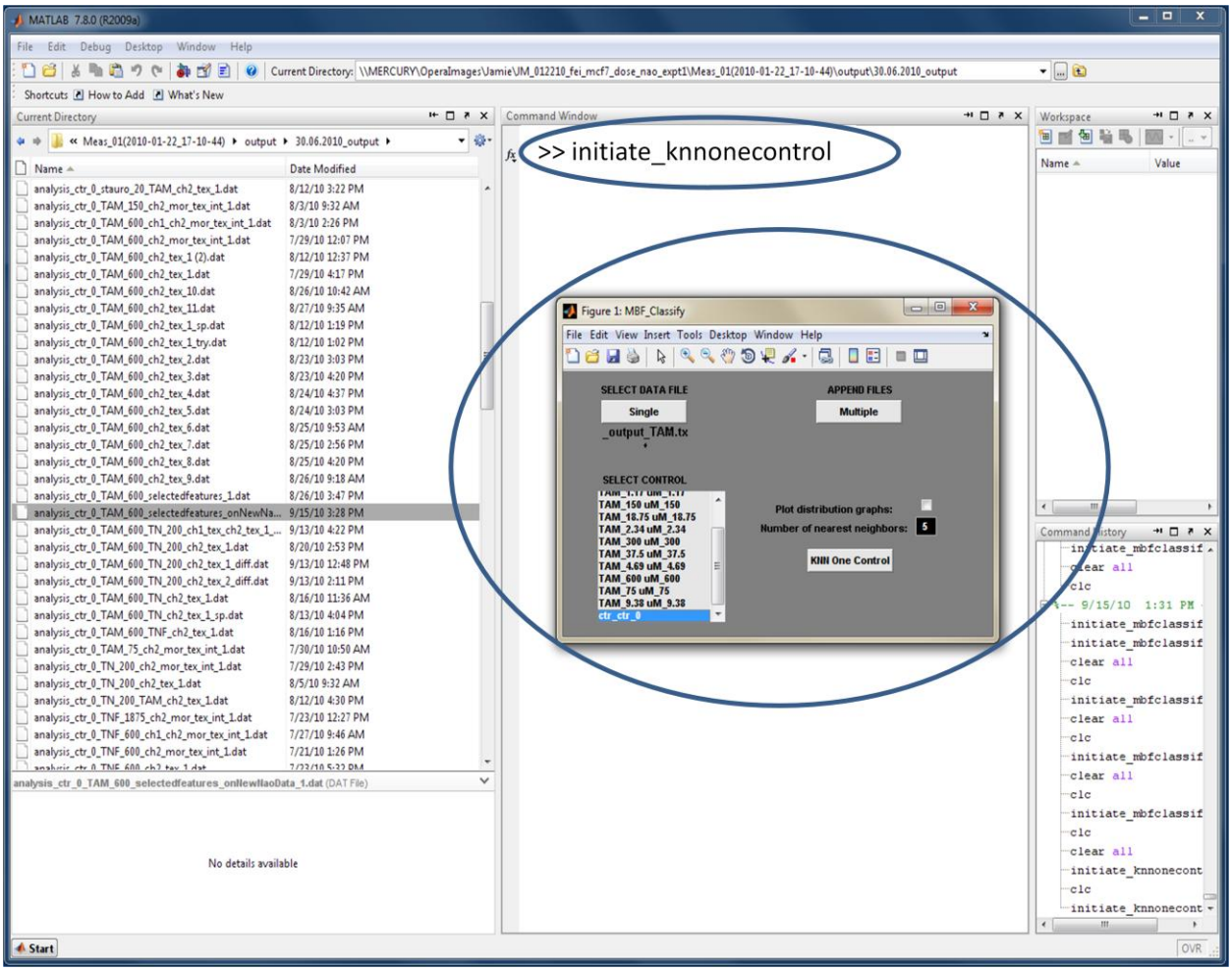
A.1 KNN SINGLE CONTROL ALGORITHM

While working with high content screening data, there can be situations when only a single control is present to create a classifier. For example, the single control can be the set of objects that were not affected by a particular treatment and hence, can be called a negative control. In order to proceed towards classification using MBF_Classify, there is a requirement of at least 2 controls. Hence, software called KNN single control was designed that performs a comparison between the single control, usually the unaffected objects and all the other objects in the population to pick those objects as the second control that are most distinct from the unaffected population. This is done by comparing the distances from the unaffected population to a benchmark to the distances of a given query to the benchmark using the KS test. The classification procedure described above often fails if more than 50% of the „treated control" cells were unaffected. In this situation the „treated control" is not really an appropriate control. To create a more useful control set we created the KNN single control algorithm. Using this algorithm the user selects from the treated cells those that are significantly (we usually use p = 0.1) different than the normal cells. This group of cells is then used as the positive control in the classifier. The alternative, and what other software programs do, is to let the user manually select positives based on visual inspection. At the moment we do not favor this approach but if you want to use it there is a way to do it. To manually select positive controls one selects them using Acapella and then uses the feature extraction script to extract the features from the selected cells. These are then provided to MBF_classify as a positive control set.

A.2 PROTOCOL

The KNN one control algorithm can be launched through MATLAB in a similar way as described for the other user interfaces above.

1) Type ***"initiate_knnonecontrol"*** on the command prompt to launch the interface. The figure below shows the command to launch the interface along with the interface.



2) Once the interface opens, the user can select a single file to upload by clicking in the ***"single"*** button or upload multiple files that would be appended together by hitting the ***"multiple"*** button. The name of the file appears on the interface once it is done uploading it.
3) The next step is to select the single control from the list that appears in the select control column. The list appears automatically once the upload of the file is complete.

   NOTE: That while multiple sets of data can be analyzed to generate a control set only a single control can be selected from the list for each analysis.
4) The user then has the option of either plotting the distributions of the distances of the control and the samples from the benchmarks (if you want to visually determine how overlapping the distributions are) or directly starting the analysis by clicking on the ***"KNN one control"*** button.

   NOTE: Since KNN computes an average distance of K number of nearest neighbors to the benchmark object, there has been included an option to specify the number of nearest neighbors that should be used for the analysis by the user.

5) A new pop-up box appears in which the user enters the p value for the analysis (the default is 0.1). In practice we have found 0.1 the best but values between 0.05 and 0.5 all work to varying degrees.
6) In performing the analysis the program analyzes the untreated cells and determines the distribution for all of the cells based on all of the features (it uses all the features in the data files). It then uses a random set of cells from the untreated control as a benchmark.

In the next step the program measures the distance of all of the objects (cells) in the treated samples from the benchmark. Once the analysis is complete, the KNN one control algorithm creates as many ***"control/sample… .mat"*** files as there are treatments present in the ***"Select Control"*** column on the interface. The ***"control/sample… .mat"*** files contain the information of the objects picked as control as specified by the user and the objects (cells) in the treatments that are scored as affected by comparing to the p value selected above (usually 0.1). These objects can therefore be used as the second control to perform analysis in MBF_Classify by simply uploading the ***"control/sample_… .mat"*** files on the interface. If there are multiple .mat files they can be appended to each other in the main part of MBF_classify.

APPENDIX (B)

B.1. DATA FLOW THROUGH MBF_CLASSIFY

The figures below explain the flow of data during the process of feature reduction and supervised classification in MBF_Classify script. Random samples of equal sizes are picked and tested for each combination of feature reduction method and classification method to find the best set up.

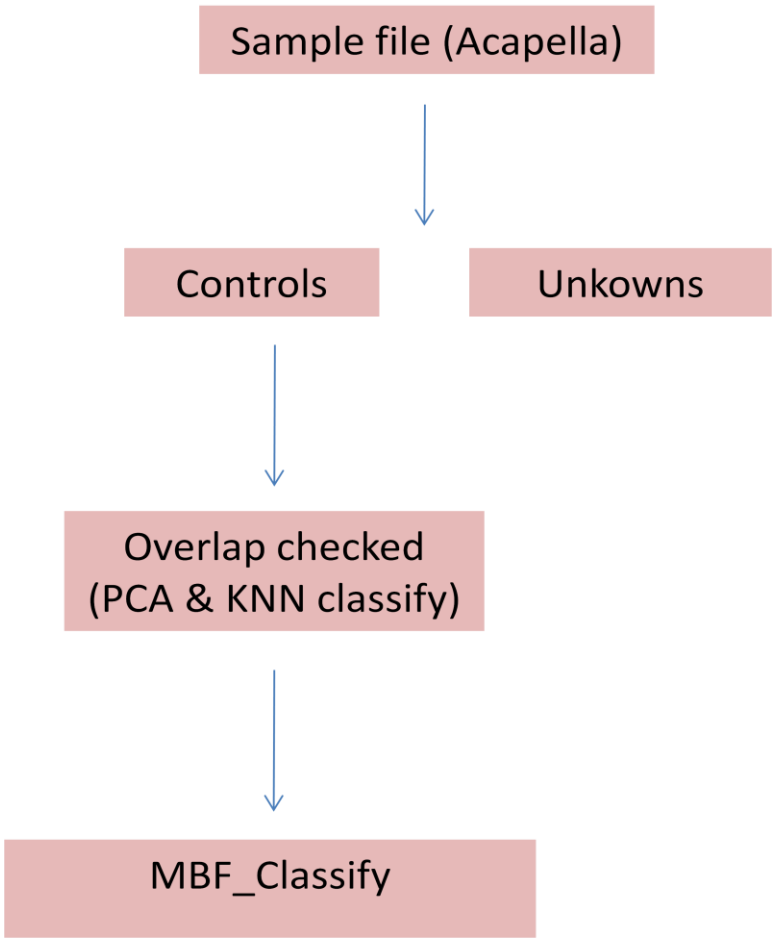
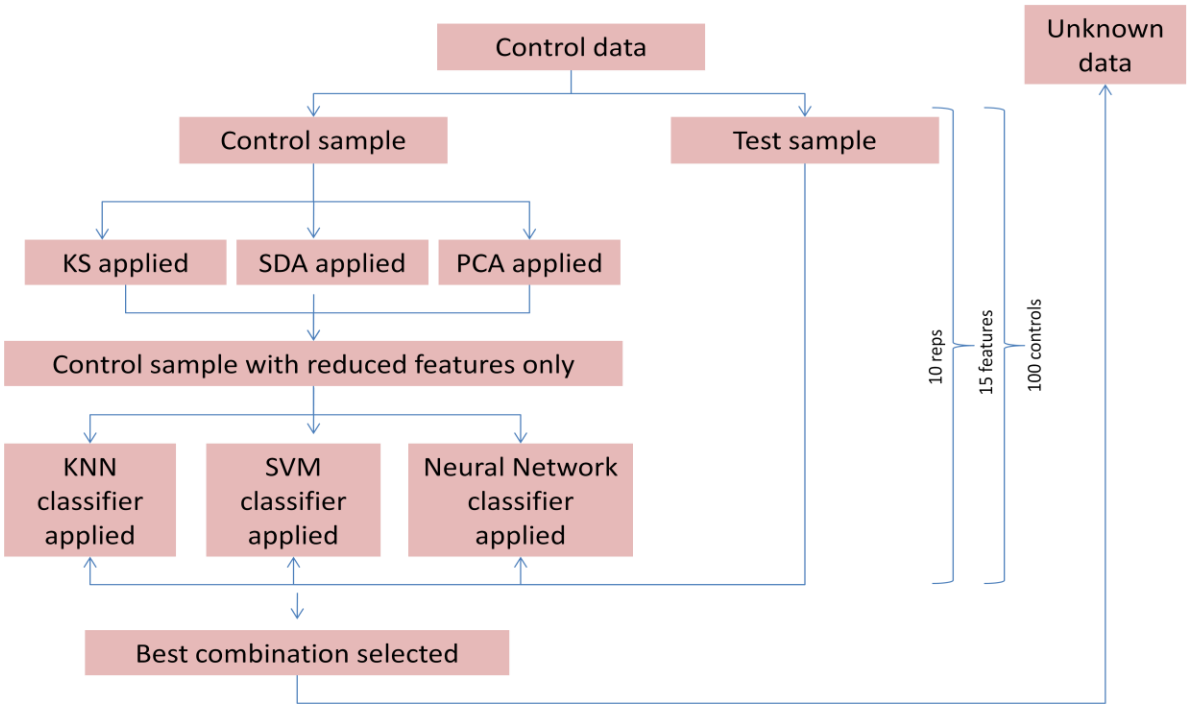
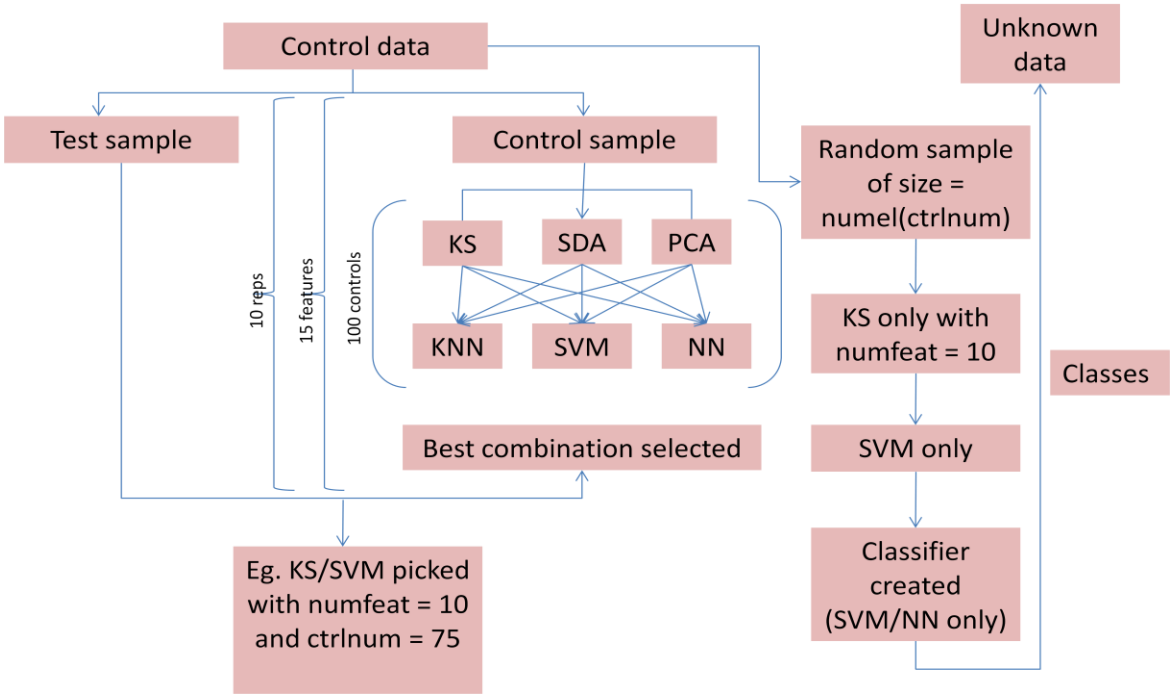


Figure 1: Data break up before MBF_Classify



(a)



(b)

Figure 2(a,b): Data flow within MBF_Classify

B.2. STATISTICAL ANALYSIS STEPS TO FIND THE BEST SET UP

In order to select the best combination of feature reduction method and classification method, the statistical steps followed are shown below. The values used here represent actual values from an analysis performed using the default values of number of features (15), number of controls (100) and number of repititions (10).

Step 1

Example: Numbers from statistical analysis to chose the best combination

ctrlvect =

100
100
100
100
100
100
100
100
100

featvect =

1	2	3	4	5

infoset =

'knn3'	'ks'
'knn3'	'sda'
'knn3'	'pca'
'svm012'	'ks'
'svm012'	'sda'
'svm012'	'pca'
'network'	'ks'
'network'	'sda'
'network'	'pca'

allmeans =

76.3033	77.1327	77.8041	77.2907	76.1058
72.1959	77.9226	76.3428	75.3160	77.6066
62.5987	63.6651	68.8784	72.3144	70.8136
75.3160	79.4629	78.0806	77.1327	78.8705
76.9747	75.2765	81.3586	74.6445	80.7662
50.7109	63.6256	68.6414	71.1295	74.4076
78.7915	73.4530	76.9547	72.2113	76.0702
73.3412	77.7515	80.9347	80.0834	74.2552
61.6039	62.3478	61.8882	72.2644	74.3673

allerrors =

1.8508	0.9698	1.6602	0.8322	3.0400
4.3215	2.0431	3.0493	5.3519	2.3785
5.2137	2.6290	6.3721	2.4176	3.8069
1.6346	0.5845	2.1943	2.3785	0.6080
3.0122	3.8581	3.5174	5.0865	6.6587
1.9720	1.0123	3.6717	4.6077	2.0556
1.1669	1.8892	2.3668	9.8366	2.2630
6.1234	2.0988	3.0622	1.5610	4.2877
0.0000	5.6780	13.9695	4.4771	4.7980

allnums =

1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	1.6667	2.3333	2.3333	2.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	1.3333	2.6667	2.6667	3.3333
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	1.6667	2.6667	3.6667	2.6667
1.0000	2.0000	3.0000	4.0000	5.0000

- allmeans: means of overall accuracy over 10 repetitions for each number of features
- allerrors: standard deviation of overall accuracy over 10 repetitions for each number of features
- allnums: means of the number of features used over 10 repetitions for each number of features

Step 2

Example: statistics

- Remove allmeans with allerrors < 0.01

featvect =

1 2 3 4 5

infoset =

'knn3' 'ks'
'knn3' 'sda'
'knn3' 'pca'
'svm012' 'ks'
'svm012' 'sda'
'svm012' 'pca'
'network' 'ks'
'network' 'sda'
'network' 'pca'

allmeans = Mean over 10 reps

72.4724	73.6177	76.8167	75.2370	74.4471
73.4202	74.6840	73.4202	76.1058	76.5008
57.5829	60.5055	71.6825	74.5261	71.2875
75.0790	78.1596	75.4344	79.6209	78.9889
74.8420	76.3428	80.0158	79.3839	80.3318
59.4787	67.4566	71.9589	74.7235	76.0664
76.7378	76.5798	75.8407	74.7566	75.0768
72.2447	74.5325	72.4531	76.3579	79.0410
70.3791	64.7098	70.6954	76.9209	74.3748

allerrors = Standard error over 10 reps

1.7143	4.5771	0.7617	0.6597	1.1009
2.7770	2.2050	5.2272	4.8626	3.0539
4.8259	4.0981	2.0658	0.8544	3.0722
3.8013	0.6841	1.6602	2.1426	0.5845
4.3410	3.5983	1.1508	3.0231	2.3667
10.8921	5.5096	0.6737	7.5721	2.4626
1.0751	4.0981	3.0678	6.6676	2.8958
0.1790	2.3865	13.7952	3.8913	2.1745
0	1.2336	6.0411	0.9137	6.0862

allmeans =

72.4724	73.6177	76.8167	75.2370	74.4471
73.4202	74.6840	73.4202	76.1058	76.5008
57.5829	60.5055	71.6825	74.5261	71.2875
75.0790	78.1596	75.4344	79.6209	78.9889
74.8420	76.3428	80.0158	79.3839	80.3318
59.4787	67.4566	71.9589	74.7235	76.0664
76.7378	76.5798	75.8407	74.7566	75.0768
72.2447	74.5325	72.4531	76.3579	79.0410
0	64.7098	70.6954	76.9209	74.3748

Step 3

Example: Statistics

- Remove allmeans for which the average number of features used is less than 90% of the number of features that is required

featvect =

1 2 3 4 5

infoset =

'knn3' 'ks'
'knn3' 'sda'
'knn3' 'pca'
'svm012' 'ks'
'svm012' 'sda'
'svm012' 'pca'
'network' 'ks'
'network' 'sda'
'network' 'pca'

allnums =

1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	1.6667	2.0000	1.6667	2.3333
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	2.3333	2.0000	2.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	2.6667	2.0000
1.0000	2.0000	3.0000	4.0000	5.0000

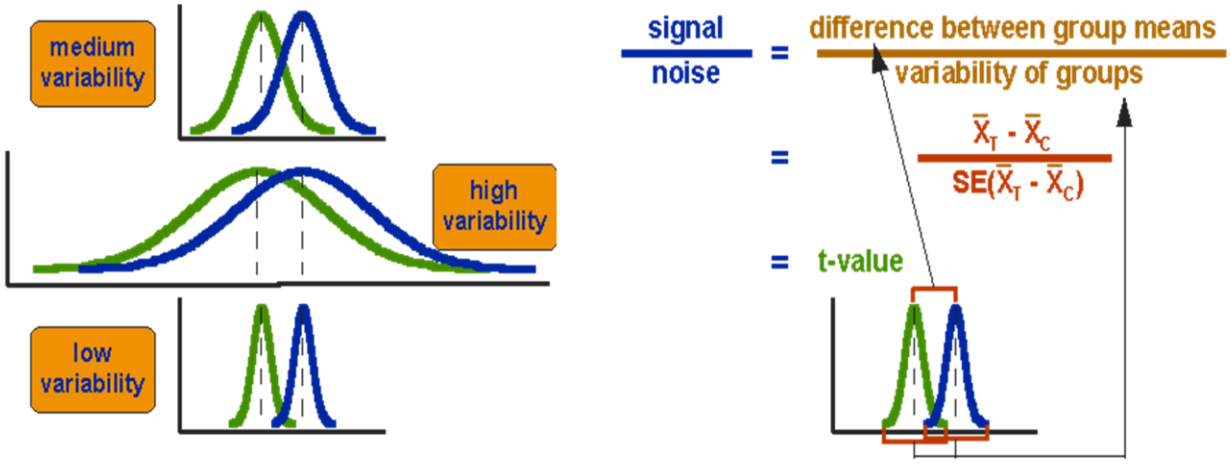
allmeans =

72.4724	73.6177	76.8167	75.2370	74.4471
73.4202	0	73.4202	0	0
57.5829	60.5055	71.6825	74.5261	71.2875
75.0790	78.1596	75.4344	79.6209	78.9889
74.8420	76.3428	0	79.3839	80.3318
59.4787	67.4566	71.9589	74.7235	76.0664
76.7378	76.5798	75.8407	74.7566	75.0768
72.2447	74.5325	72.4531	0	79.0410
0	64.7098	70.6954	76.9209	74.3748

Max of allmeans: Row 5, Column 5 → **Two sample t-test**

Step 4

Two sample t-test



Ho (Null Hypothesis): Samples come from populations with statistically equal means
H1 (Alternate Hypothesis): Samples come from populations with statistically different means
Significance level : 0.05

Step 5

```
i = allmeans(rows)
j = allmeans(columns)

allmeans =

  72.4724  73.6177  76.8167  75.2370  74.4471
  73.4202   0  73.4202   0   0
  57.5829  60.5055  71.6825  74.5261  71.2875
  75.0790  78.1596  75.4344  79.6209  78.9889
  74.8420  76.3428   0  79.3839  80.3318
  59.4787  67.4566  71.9589  74.7235  76.0664
  76.7378  76.5798  75.8407  74.7566  75.0768
  72.2447  74.5325  72.4531   0  79.0410
     0  64.7098  70.6954  76.9209  74.3748
```

```
allerrors =
   1.8508   0.9698   1.6602   0.8322   3.0400
   4.3215   2.0431   3.0493   5.3519   2.3785
   5.2137   2.6290   6.3721   2.4176   3.8069
   1.6346   0.5845   2.1943   2.3785   0.6080
   3.0122   3.8581   3.5174   5.0865   6.6587
   1.9720   1.0123   3.6717   4.6077   2.0556
   1.1669   1.8892   2.3668   9.8366   2.2630
   6.1234   2.0988   3.0622   1.5610   4.2877
   0.0000   5.6780  13.9695   4.4771   4.7980
```

Two sample t test:
Population 1 : Population corresponding to maximum mean i.e. row 5, column 5
Population 2: Population corresponding to all the other means in the matrix i.e. a loop for (i,j)

H(i,j) obtained

```
H =

   1   1   1   1   1
   1   1   1   1   1
   1   1   1   1   1
   1   1   1   0   0
   1   1   1   0   0
   1   1   1   1   1
   1   1   1   1   1
   1   1   0   1   0
   1   1   1   1   1
```

```
infoset =
    'knn3'     'ks'
    'knn3'     'sda'
    'knn3'     'pca'
    'svm012'   'ks'
    'svm012'   'sda'
    'svm012'   'pca'
    'network'  'ks'
    'network'  'sda'
    'network'  'pca'
```

```
featvect =
     1     2     3     4     5
```

```
allnums =

   1.0000   2.0000   3.0000   4.0000   5.0000
   1.0000   1.6667   2.0000   1.6667   2.3333
   1.0000   2.0000   3.0000   4.0000   5.0000
   1.0000   2.0000   3.0000   4.0000   5.0000
   1.0000   2.0000   2.3333   2.0000   2.0000
   1.0000   2.0000   3.0000   4.0000   5.0000
   1.0000   2.0000   3.0000   4.0000   5.0000
   1.0000   2.0000   3.0000   2.6667   2.0000
   1.0000   2.0000   3.0000   4.0000   5.0000
```

Step 6

Finding best set up

whichone.rowheaders =		whichone.means =	whichone.stds =	whichone.ctrlnums =	whichone.numfeat =	whichone.nums =
'network'	'sda'	72.4531	13.7952	100	3	2.6667
'svm012'	'ks'	79.6209	2.1426	100	4	4.0000
'svm012'	'sda'	79.3839	3.0231	100	4	2.6667
'svm012'	'ks'	78.9889	0.5845	100	5	5.0000
'svm012'	'sda'	80.3318	2.3667	100	5	3.3333
'network'	'sda'	79.0410	2.1745	100	5	2.6667

Where,

whichone.numfeat is extracted from featvect and gives the number of features required to be used

whichone.nums is extracted from allnums and gives the number of features actually used

featvect =

1 2 3 4 5

allnums =

1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	1.6667	2.0000	1.6667	2.3333
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	2.3333	2.0000	2.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	4.0000	5.0000
1.0000	2.0000	3.0000	2.6667	2.0000
1.0000	2.0000	3.0000	4.0000	5.0000

Step 7

Best choice is the one with the least standard deviation

Result:

Bestchoice.std = 0.5845
Bestchoice.mean = 78.9889
Bestchoice.ctrlnum = 100
Bestchoice.numfeat = 5
Bestchoice.num = 5
Bestchoice.frmethod = ‘ks’
Bestchoice.classifier = ‘svm012’