CellClassifier User Manual

Contents

Getting started

What is CellClassifier?

CellClassifier is a visual tool to create training sets of cellular phenotypes. These training sets are then used to train a Support Vector Machine based classifier. The classifier can classify all the cells in an assay into the user defined phenotypical classes. The tool runs on Matlab (developed with Matlab version R2008a) and requires that the images are first analyzed with CellProfiler (developed with CellProfiler version 1.0.5810) and that the Statistical Pattern Recognition Toolbox (stprtool) machine learning package (developed with stprtool version 1.0.4) is installed.

LINKS

CellProfiler: http://www.cellprofiler.org/
Matlab : http://www.mathworks.com/
Stprtool: http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html

Installation

Copy the files CellClassifier.m and CellClassifier.fig into a directory of your choice. Add this directory to the path of Matlab. You can run the tool in Matlab by typing "CellClassifier". For full functionality you must first install the Stprtool. Please refer to the User's Manual of stprtool how to install the package.

CellProfiler

Before you can do anything with CellClassifier you have to first analyze your images with CellProfiler. CellProfiler is tool that does cell segmentation and feature extraction on microscope images. Please refer to the CellProfiler User Manual. CellClassifier requires that at least the primary objects are detected from the images and at least one feature extraction module has been used.

First run

After you have managed to analyze your images with CellProfiler, CellProfiler saves an output file with all the analysis results. Open CellClassifier and create a new project with "File-New". A directory listing opens and you can select the CellProfiler output file. Now the data and images are loaded into CellClassifier. Most likely you will have to rescale the colors with "View-Rescale Colors" tool in order to see the images properly on the tool. Now you can continue using CellClassifier. Please read the following sections carefully for further information, especially the "An example workflow" -section.

WARNING

Do not change the location, rename, or delete the original images, since they are required by CellClassifier.

Supervised Clustering of Cellular Phenotypes

Supervised clustering

Clustering algorithms can be roughly divided into two main groups: unsupervised and supervised clustering. In unsupervised clustering a machine learning algorithm groups the data points without

any insight given by the user. On the contrary, in supervised clustering the user gives examples of objects (the training set) that belong to different user defined classes and the machine learning algorithm tries to find the rule in the multidimensional feature space how to separate these objects from each other as well as possible. Many algorithms have been proposed for both types of machine learning approaches over the years. CellClassifier uses Support Vector Machines (SVM's). SVM's have been proven to be a powerful algorithm for supervised clustering. They tend not to over fit to the training data and we believe that they are especially suitable for the "iterative machine learning" approach explained in the following. In addition, there are several free software packages that provide readily usable code for SVM's. And finally, modern SVM's support the multiclass supervised clustering.

Iterative machine learning

In a classical supervised clustering setting the machine learning algorithm is given a training set of objects and the algorithm is trained with this set of data. The clustering efficiency can then be tested with a separate test set of objects that the user has classified, but which is not shown to the SVM algorithm before training. If the efficiency is satisfactory the SVM algorithm is run over the full set of data and all the object are classified.

In iterative machine learning we repeat the process of training and testing several times. At the first round the user gives examples of objects belonging to some classes and the machine learning algorithm is trained with this data. In the second round, the algorithm shows examples of objects it thinks that belong to these classes. Now, the user merely adds objects to the improved training set which the machine learning algorithm has put into a wrong class. That is, the user only corrects the "misunderstandings" of the algorithm. In this way we can concentrate on difficult examples of objects that are hard to classify or are for some reason easily missed by humans. Such objects may lie close to the decision boundaries or in the periphery in the multidimensional feature space. This iterative process is continued until the machine learning algorithm does not make any mistakes or the classification results do not improve anymore.

CellClassifier is designed to fully support the iterative machine learning approach.

Basic Functions of CellClassifier

When first opened the CellClassifier shows the first image of the dataset. Each detected cell is marked with a dot in the middle of the cell. In CellClassifier all the marked objects are clickable. This means, that the user can assign a class to an object simply by clicking on it.

Full view

Full view is the standard way in CellClassifier to show the images. In this view the whole image and all or some of the corresponding channels are displayed in the main window of the tool. The detected cells are marked with a dot in the middle and also the object outlines are drawn, if they are available (and this option is activated in the Settings). The full view allows the user to see the cellular context where each cell is.

Panels

In the panel view individual objects are cropped from the images and organized on a lattice. In this way the user can display many cells side by side for comparison and, for example, easily detect cells that are classified into a wrong class.

Output file format

The saved file includes all the settings that CellClassifier uses. In addition, the file includes the classes for individual cells that the user or CellClassifier has classified. The cell based data is important for advanced users who want to access the classification of individual cells. For typical

users it is easier to import the data to an Excel sheet using the “File-Export” function. The excel sheet contains the total number of phenotype cells in each class and type. The Excel sheet does not contain classification information of individual cells.

The output file is a Matlab file that contains a struct object of the name “Save_File”. The object contains the following fields:

Settings: All the settings of CellClassifier are saved here. A typical user might only need the field Save_File.Settings.Class_Names that gives the names of the classes. The position of the name corresponds to the index of the class.

Trained_Cells: Cell array of vectors. Each image in the data has one cell. The vector is of length how many objects are detected in the image. The object that the user has clicked are given the class number. The image and object order is the same as in the CellProfiler output file.

Classified_Cells: Cell array of vectors. Each image in the data has one cell. The vector is of length how many objects are detected in the image. The objects are given the classified class. The image and object order is the same as in the CellProfiler output file.

An example workflow

A typical work flow is the following.

- Open a CellProfiler output file with “File-New”
  This might take a long time if the output file is very big. Sometimes the output file is so big that is does not fit the computer memory. In this case we recommend to divide the image set into pieces and analyze each of them separately with CellProfiler.
- Rescale the colors with the “View-Rescale colors” tool
- Add classes with “Classes-Add” (at least two of them)
- Select an active class with the “Classes” menu or the mouse wheel
- Click on cells that belong to the active class. Repeat for all classes
- Select measurements from “Classifier-Measurement Setup”
  Try to select all measurements that seem relevant to the phenotype of interest. In general, it is better to have too many features than too few. The SVM algorithm is rather robust to uninformative features and usually does not overfit to the training data too much. However, overfitting is possible and by reducing the number of features you might get better classification results.
- Normalize the measurements with “Classifier-Normalize” button
  Z-scoring to All works in most cases well
- Train the SVM with the training data by clicking “Classifier-Train”
- Check the training quality with “Classifier-Confusion Matrix”
- Classify all images in the data with Classifier-Classify All
  This may take a long time if the data set is big.
- Go through images with “Classifier-Show Classified” on and correct wrongly classified cells.
  Hint. Go to the images with the highest number of a chosen phenotype with the “View-Go to Image”. This way you can easily detect false positives.
- Repeat from training until you are satisfied with the classification quality
- Go to the panel view with “View-Go to Panel” and correct misclassified cells.
  The panel view might have empty slots if the class is rare
- Classify for the last time with “Classifier-Classify All”
- Save the data with “File-Save”
- Export results to an excel sheet with “File-Export Data”

Menus

File Menu

With this menu you can create, load and save CellClassifier projects, export classification data, save screenshot images, change settings, parse information from the image file names, show the about screen, and exit CellClassifier.

New

Creates a new CellClassifier project. You must select the CellClassifier output file that you want to analyze with CellClassifier. The CellClassifier memory is emptied and a new CellClassifier project with the selected data is created.

Load

Loads and restores a saved CellClassifier project. All the CellClassifier settings are restored from the file.

Save

Saves the CellClassifier project into a file. (For expert users: single cell data can be extracted from this file.)

Export Data

Saves the classification results into an Excel sheet. You have to have first created training data, trained the SVM, and classified all images before this option is activated. The Excel contains the number of cells in each class in the selected type. The target type can be selected from the "File-Settings-Combine Export Data to" option. Possible entities are All, Image, and all the other types created by "File-Parse Images". Typical types can be for example, image, well, plate etc.

Export Image

Saves the currently active view as an 8bit PNG image. The numbers and cell middle points are not saved into the image. The outlines are saved if they are currently active.

Settings

Opens a Settings window. You can change the following options:

Combine export data to

Selects the type to which the number of cells in a class are summed up to.

Show middle points

Selects whether the cell middle points are showed on the screen or not.

Show segmentation

Selects whether the outlines are showed on the screen or not. Requires that the outlines are saved by the SaveObjectSegmentation CellProfiler module.

Panel resolution X

Selects the horizontal resolution for the Panel view. The bigger the resolution is the more you can have cropped cells in the panel. With a higher resolution you can better zoom into the image.

Panel resolution Y

Selects the vertical resolution for the Panel view. The bigger the resolution is the more you can have cropped cells in the panel. With a higher resolution you can better zoom into the image.

Panel box size

Selects the size of the cropped individual cell images in the panel view. The cropped images are squares.

Advanced Settings

Opens a new settings window where you can change the advanced settings. These settings mostly change how the machine learning is done. The default settings should be adequate for most users. In order to get the best possible classification results with your data you might need to change some of the settings.

Classificator function

Selects between which machine learning algorithm is used for classification. Default options are SVM, Multilinear perceptrons, and K-nearest neighbors (KNN) algorithms. In addition, the choice Custom is available which allows to use any other machine learning algorithm. The SVM parameters can be changed by the advanced settings given below. If Multilinear perceptron or KNN is chosen the following SVM parameters are not used. The default parameter for KNN is K=8.

To use a custom algorithm open CellClassifier.m in an editor. Search for word “Custom” (code must be edited in 4 different places). How to add your custom functions is explained in the code.

SVM Kernel function

Selects which kernel function is used in the SVM. Possible choices are “Radial Basis Function (RBF)” (default), “Linear”, “Polynomial”, and “Sigmoid”. Please refer to the stprtool User’s Manual for further information.

SVM Parameter arg

Selects which argument value is used in the SVM. Please refer to the stprtool User’s Manual for further information.

SVM Parameter C

Selects which parameter value C is used in the SVM. Please refer to the stprtool User’s Manual for further information.

SVM Parameter tmax

Selects the maximum number of iterations for the SVM optimization. The “Infinite” option sets no upper limit. Please refer to the stprtool User’s Manual for further information.

SVM Verbose

Selects whether extra information about the SVM training is displayed on the screen or not. Please refer to the stprtool User’s Manual for further information.

Feature set minimization

Selects the feature set minimization method. Possible choices are “None” (default), ‘’Principal component analysis PCA)” and “Fisher’s Linear Discriminant Analysis (LDA)”. PCA performs the optimal linear transformation on the original data matrix that keeps the most of the variation in the data. LDA performs the linear transformation that optimally separates the objects in different classes in the training set.

Number of features

Selects the number of features after PCA or LDA transformation.

Parse Images

Parses useful information from the image file names. These information can be for example well or plate name or position etc.

- "Use automatic well detection" works for most standard microscope naming standards. By activating it and pressing Ok the well name, row, and column information are extracted from every image.
- Well to gene name mapping file. If a type called Well_Name is parsed (done automatically by "automatic well detection"), extra meta information about the well, for example the gene name corresponding to the well, can be added to the images. Each image originating from the same well is given the same meta information. By clicking the browse button you can select an Excel sheet that has two columns. The first row must be the following: "Well, new_name", where new_name is given by the user. Each row below has the well name in the first column (e.g. A01) and the value of the type on the second column.
- Advanced parsing. With this tool the user can parse the image file names with any regular expression.

About

Displays a window that gives the copyright information of CellClassifier.

Exit

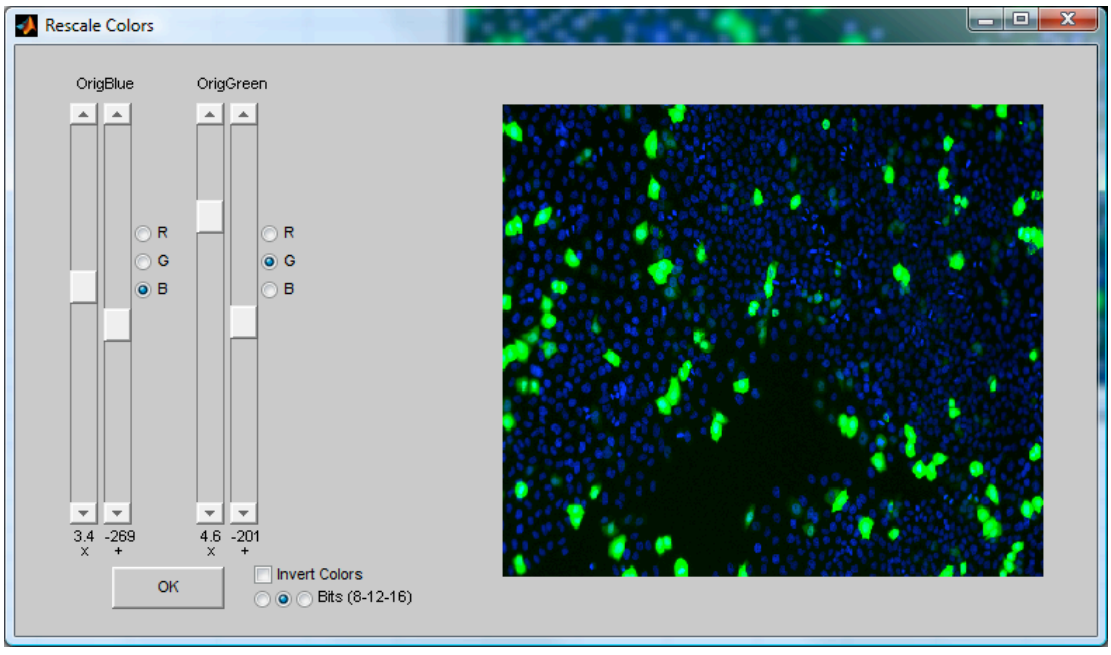
Exits CellClassifier.

View Menu

With this menu you can change and adjust the colors of the images and move to different images and to the panel view.

Rescale Colors

Opens the color rescaling tool.



Each available channel has two slide bars and the name of the channel is on top of the bars. The first slidebar multiplies the color values in the corresponding channel with the chosen value (i.e. changes the contrast). The second slidebar adds the chosen constant to the color values in the corresponding channel (i.e. sets the level for black). With these two settings you can set the color

rescaling so that the background is completely black and the interesting signal uses the full intensity range of the image.

With the three radiobuttons you can assign a color for each channel. The possible choices are nothing: this channel is not shown, R:red, G:green, B:blue, R+G: yellow, R+B: magenta, G+B: cyan, and R+G+B: white. In total you can have 7 channels displayed at the same time with different colors. Many phenotypes are easier to see by eye with non-classical color combinations (for example 1st channel with cyan and the 2nd channel with yellow etc). Experiment yourself to find the best possible colors for your images. You can invert all colors with the Invert Colors button and choose the correct image bit depth with the Bits (8-12-16) buttons.

The small image shows a preview of the final color settings. By clicking OK the settings are saved and CellClassifier starts to use the new colors.

Next Image

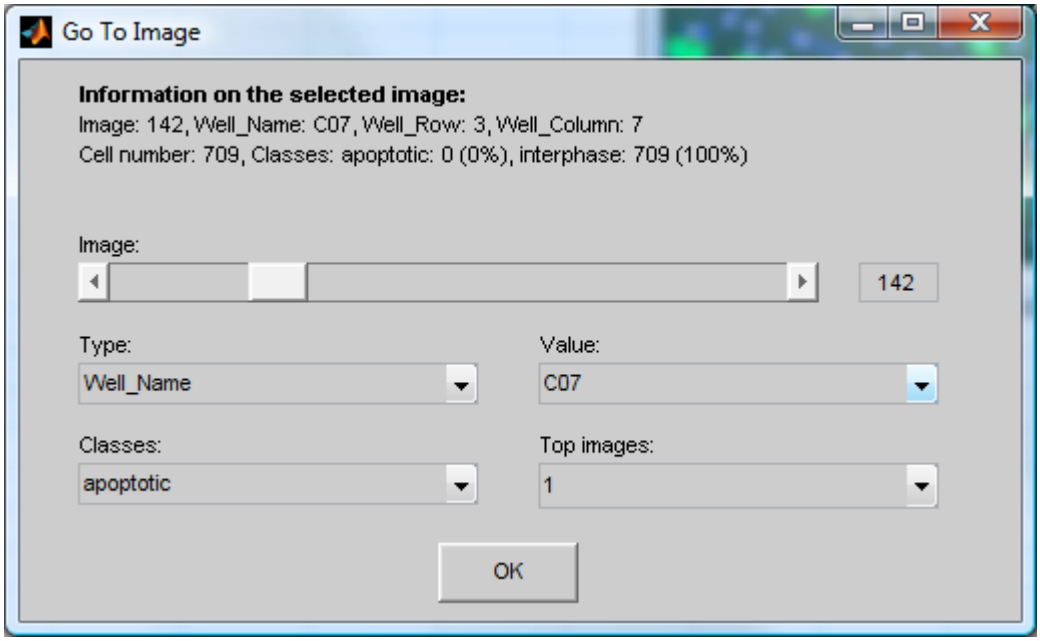
Shows the next image. The image order is defined by CellProfiler and ultimately by the microscope.

Previous Image

Shows the previous image. The image order is defined by CellProfiler and ultimately by the microscope.

Go to Image

Opens the image browser tool.



With this tool you can go directly to a preferred image. With the “Image” slide bar or by changing the number beside the slide bar you can directly go to a certain image number. You can select the type from the “Type” drop-down menu and go to an image with the selected type and value by choosing a value from the “Value” drop-down menu. These two menus are not shown if no meta information is parsed from the image file names. By choosing a class from the “Classes” drop-down menu you can go to the image with the highest number of this phenotype class by selecting a number from the “Top images” drop-down menu. These two menus are not displayed if all images are not classified with ”Classifier-Classify All”.

On top of the window there is an information box that shows information on the current image. These are the image number, all parsed meta information for the image, total cell number, and number and percentages of cells in the classes.

Go to Panel

Shows the panel view on the main window of CellClassifier. Cells that belong to each class are gathered from many randomly chosen images. The images of cells are cropped and reorganized in a lattice in the corresponding class. The panel view requires that all the cells are first classified with "Classifier-Classify All". In the panel view cell phenotypes are easy to compare to each other. You can change the size of the panel image and the size of the cropped cell image in "File-Settings". Every time a new panel is created the cells are chosen randomly. It is not possible to go back to a certain panel once it's lost. If some classes are very rare, the same cells may be displayed several times or there are empty slots in the panel.

Classifier Menu

Show Current

When "Show Current" is selected only on the cells that the user has clicked a number is drawn.

Show Classified

When "Show Classified" is activated the SVM classified class is displayed on the cell with a different font. If the user clicks on the cells the new user given class is displayed instead. This option is activated after "Classifier-Classify All" is run.

Normalize

Normalizes the raw measurement data.

Normalize measurements

Selects the type according to which the single measurements are normalized. The default is "All". Separate normalizations are performed for every type. Depending on the application sometimes it is useful to normalize each plate or well separately.

Normalization method

Selects the normalization method. The default is "log Z-score". The possible choices are:

- log Z-score: First takes the natural logarithm of each feature from the cells that belong to the current type. Then, the mean of the feature is subtracted and then the values are divided by the standard deviation of the feature.
- log MAD: First takes the natural logarithm of each feature from the cells that belong to the current type. Then, the median of the feature is subtracted and then the values are divided by the Median Absolute Deviation (MAD) of the feature.
- Z-score: The mean of the feature is subtracted and then the values are divided by the standard deviation of the feature.
- MAD: The median of the feature is subtracted and then the values are divided by the Median Absolute Deviation (MAD) of the feature.
- log: Takes the natural logarithm of each feature from the cells that belong to the current type.

Train

Trains the SVM using all the training data from all images.

WARNING
With some SVM settings and with a large training set this might take a long time

Classify All

Classifies all cells in all images with the latest SVM.

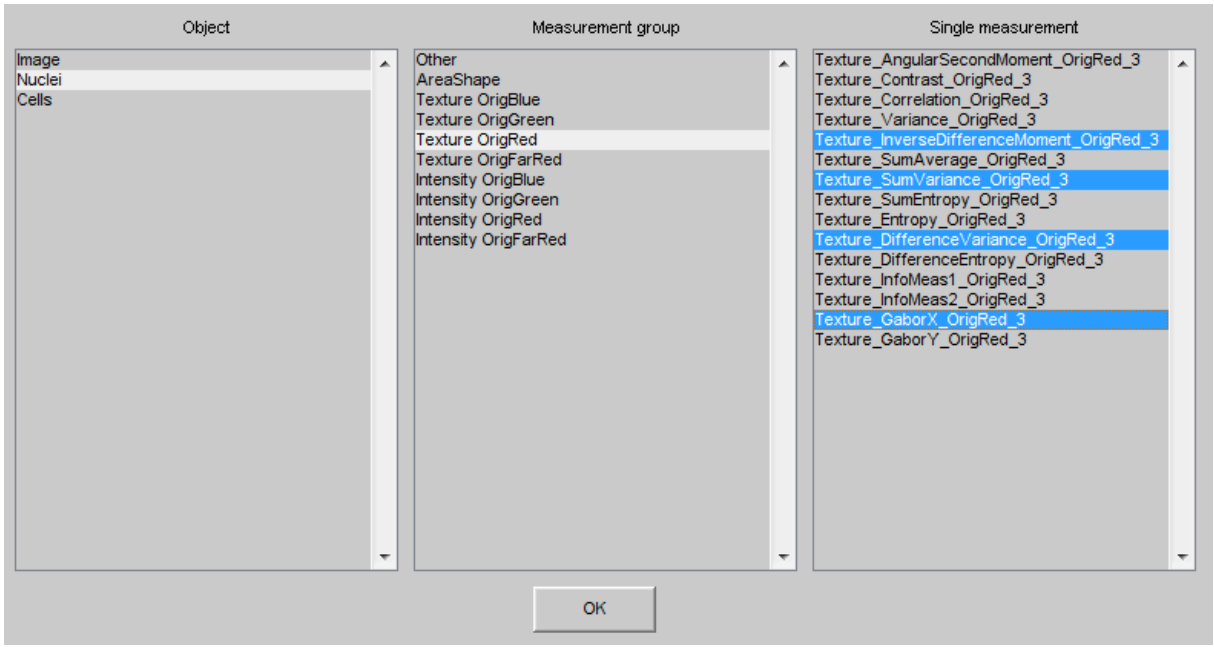
WARNING
With some SVM settings and with a large data set this might take a long time

Confusion Matrix

This button shows two confusion matrices. In the upper one all the cells in the training data are used both as the training set and the test set. The rows correspond to the user defined classes and the columns are the SVM classes. The number is the number of cells in each combination. Total gives the total number of cells in the classes. Correct % gives the percentage of correctly classified cells. The lower right percentage value is the average Correct % of all the classes. The lower confusion matrix is the same but the train and test sets are cross validated so that they have different cells. The lower confusion matrix therefore shows a more realistic results. The upper confusion matrix may suffer from over fitting. These matrices can be used to assess, for example, the quality of classification or if some classes are overrepresented in the training set compared to other classes.

Measurement Setup

Opens the measurement setup tool.



With this tool you can select which features are used. The “Object” field shows all the available object classes that are detected. The “Measurement group” shows all the measured feature groups for the chosen object. The “Single measurement” shows the single features for the chosen measurement group. You can select and deselect single features by clicking on the names of single features. The selected features are highlighted. The idea is that you choose single features that you think are relevant for the classification of the current phenotype classes. If irrelevant, too many, or too few features are selected the classification results are not optimal. In practice, the SVM algorithm is not too sensitive to extra irrelevant features so it’s better to select slightly too many features than too few. You may want to try out different selections when trying to achieve best possible classification results. By clicking OK the feature selection is taken into use.

Classes Menu

Add

Adds a new class. Type the name of the new class in the box and click OK.

Remove

Removes a class. Select the removed class from the pulldown menu. Click OK to remove the class. Cells that were trained with this class are removed from the training set. All other cells in the training set are kept.

Rename

Renames a class. Select the renamed class from the pulldown menu, write the new name in the box, and click OK.

Merge

Merges two classes into one. Select the merged classes from the pulldown menus, write the new name in the box, and click OK. The cells that were trained in with these classes are also merged together in the training set.

Unclassify

Activates the unclassify option. When you click on a cell that has been classified by the user, the class will be removed from this cell.

Class Names

Activates the chosen class. When you click on a cell it will be assigned to this class in the training set and the corresponding class number will be drawn on top of the cell.

NOTE
You may also change the class or unclassify options with the mouse wheel anywhere.

SaveObjectSegmentation CellProfiler Module

This additional CellProfiler module allows you to specify which object segmentation will be stored (in compressed PNG images) and where these images should be stored. The module has 2 user inputs:

- Select the name of the object which segmentation you would like to store. Object-names that have been defined in the current CellProfiler pipeline will appear as options in the pulldown menu.

- Enter the path name of the folder where the segmentation images will be stored. Type period (.) for default output directory. It is often good to keep the original measurement images separated from the produced object segmentation images. Leave a '.' (dot) to store the produced images in the default output directory. Note that relative pathnames can be used, for instance you could type ./AnotherSubfolder where the first period stands for the default output directory. This stores the segmentation images in the AnotherSubfolder in the default output directory. Note that the stored images are grayscale compressed PNG images, with the grayscale color corresponding to the object-index/identifier in CellProfiler.