CellProfiler Analyst Classificator Guide

V. 1

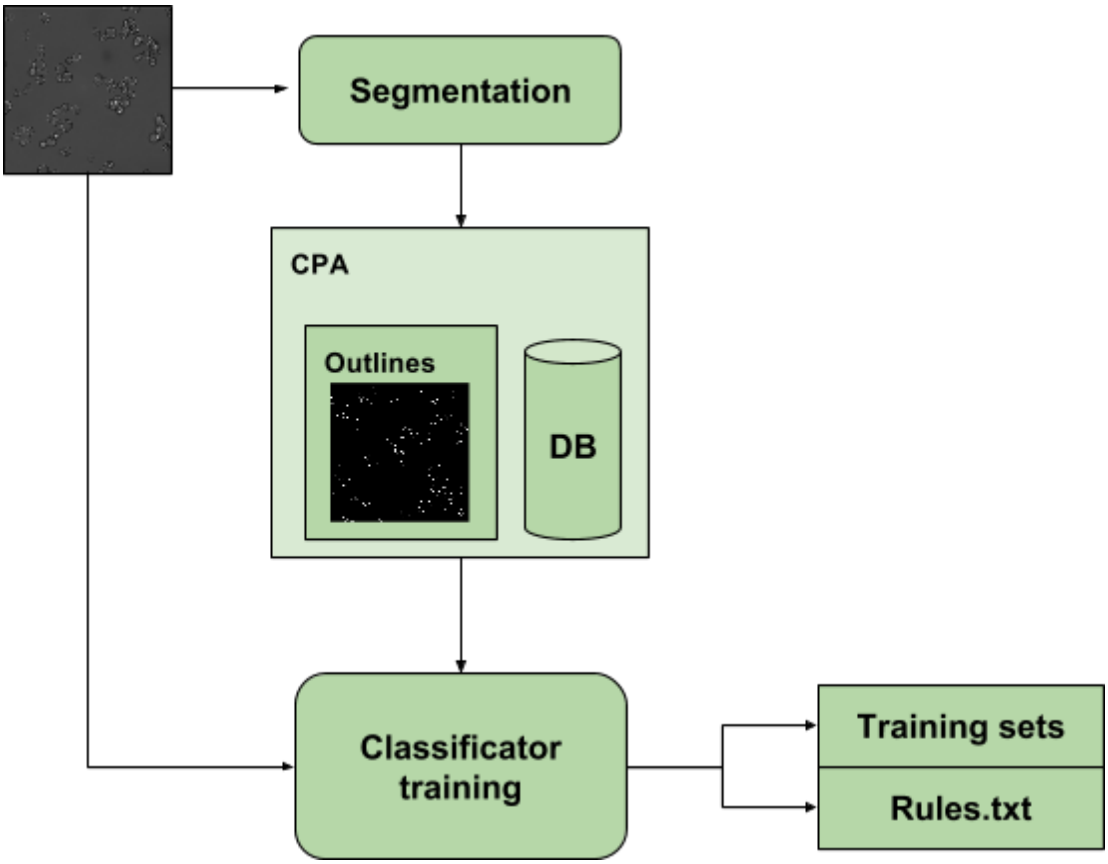
Requirements:

1. CellProfiler **2.2**
2. CellProfiler Analyst **2.2.1**
3. Python 2.7

Contents:

Overview:

The complete process consists of 2 steps (segmentation + measure + export, classification).



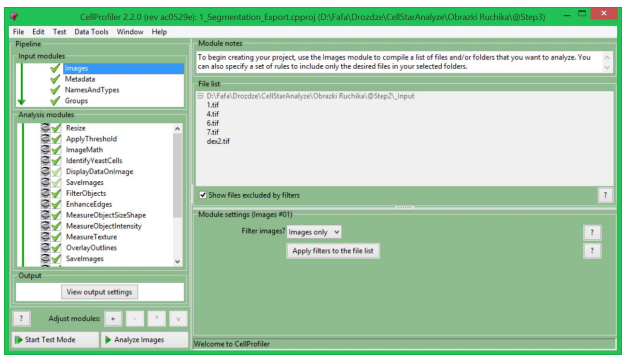
Steps:

1. Segmentation and export

This step consists of setting up and running CellProfiler 2.2 on input data, segmentation and export in CPA compliant format.

Description:

Loads images, perform segmentation and extract outlines image. Calculate measures used in classification such as intensity, shape, texture. Export all measurements to database and save input images and outlines with option to record in database.



Remarks about pipeline:

1. CPA cannot load multi-page TIF files so if it is the case then they have to resaved as single page TIF
2. Try to export one type of objects (e.g. if you prefilter then export only filtered object) otherwise CPA might be unstable.
3. If you calculate Intensity measures it is a good idea to do it on LoG (EnhanceEdges) version of input imagery as it will make sure that resulting classificator is attached to a specific lighting conditions.
4. The length of database name (internal storing) cannot be long, otherwise you will have to manually fix rules file after training as measurement names will be truncated. Length depends on image (intensity, texture) and object name so try to keep it short. For example. **PrefiltCells**_Texture_InverseDifferenceMoment_**Br_EdgedImage**_3_90 is on the brink of being truncated.

Input:

- input imagery

Output:

- cell outlines for guidance in CPA
- brightfield image for guidance in CPA (in case if multi-page TIF is input)
- YeastDB SQL-lite database for CPA
- YeastDB.* setting files for CPA

2. Classificator training

This step consists of setting up and training classifier in CellProfiler Analyst using the data collected in the previous step. See http://cellprofiler.org/cpa/5_classifier.html or watch https://www.youtube.com/watch?v=XMKgiRGb4IY for official instructions.

After training the same classifier rules can be applied to more images.

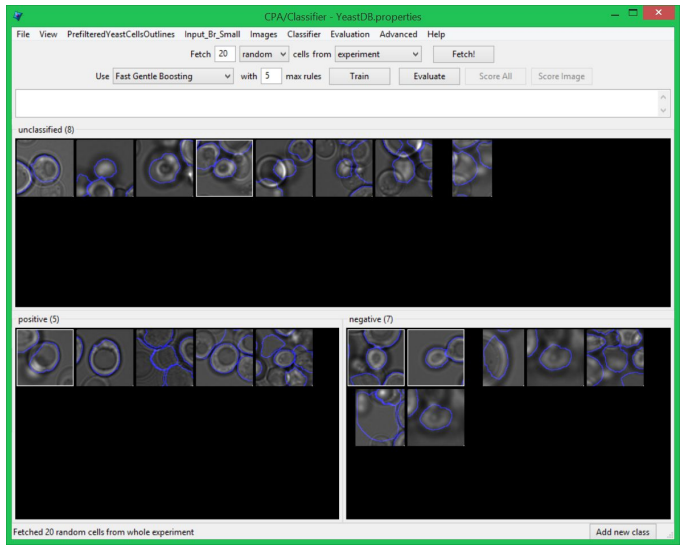
Description:

In this step the classificator which will provide specific cell filtering is trained. The resulting classificator rules in text format will be then loaded in CP. Cell outlines can be turned on and off for guidance.

Usage:

1. Setup CPA YeastDB.properties file manually or use modified CPA_Setup.py script.

The important setting to change list of columns that should ignored during training. It can prevent using measures like location to classify as it is not scalable. See more details in CPA_Setup.py.



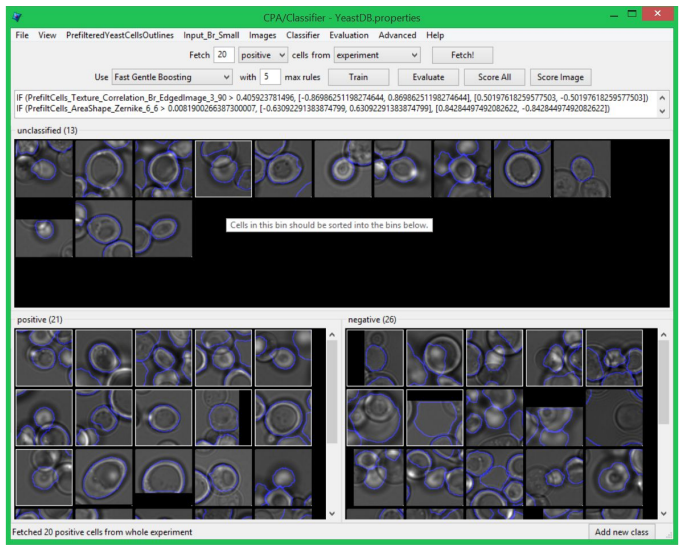
2. Run CPA and load YeastDB.properties file.
3. Train classificator (FastGentleBoosting):

The process consists of repeated fetching objects, sorting them into classes (positive, negative) and repeating until desired quality is achieved.

First switch to FastGentleBoosting classifier as it is the one compatible with CellProfiler.

The process should start with a number of random fetches of cells which we distribute to positive/negative bins using drag-and-drop. The questionable objects can be remove from the list with ‘Delete’ key.

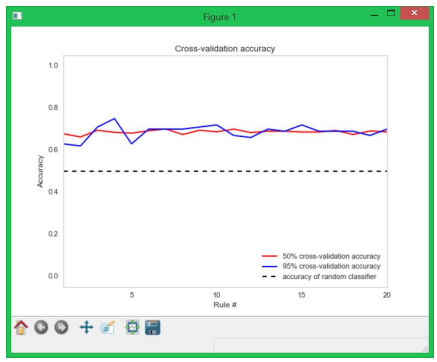
After having at least 20 samples in both bins we can Train classifier and use it to provide us with presumable positive or negative cells which allows us to quickly find and correct objects that are difficult and misclassified. Remember to **retrain** classifier before every fetch.



Training set (File->Save Training set) can be saved and loaded so the work is not lost after closing CPA.

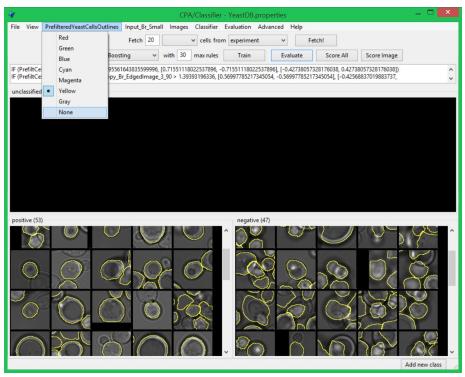
After having around 30 samples the max rules number should be set to 20 or 30 to allow classifier to be more specific.

The process should continue until the acceptable false-positive, false-negative ratio is achieved. The most accurate way to measure that is to fetch 100 positive/negative objects and count how many of them are misclassified. Another way to measure the progress of the classifier training is using **Evalute** which uses training set to retrain and test classifier and depending on the selected option in the menu show Confusion Matrix or Classification Report. In both cases the numbers will be much lower than the real accuracy as it is done on training set which consists of the most difficult cases. Plot in Classification Report show how the quality change depending on the number of rules used in two data split: 50% training / 50% testing, 95% training / 5% testing. The plot can point suggest that:

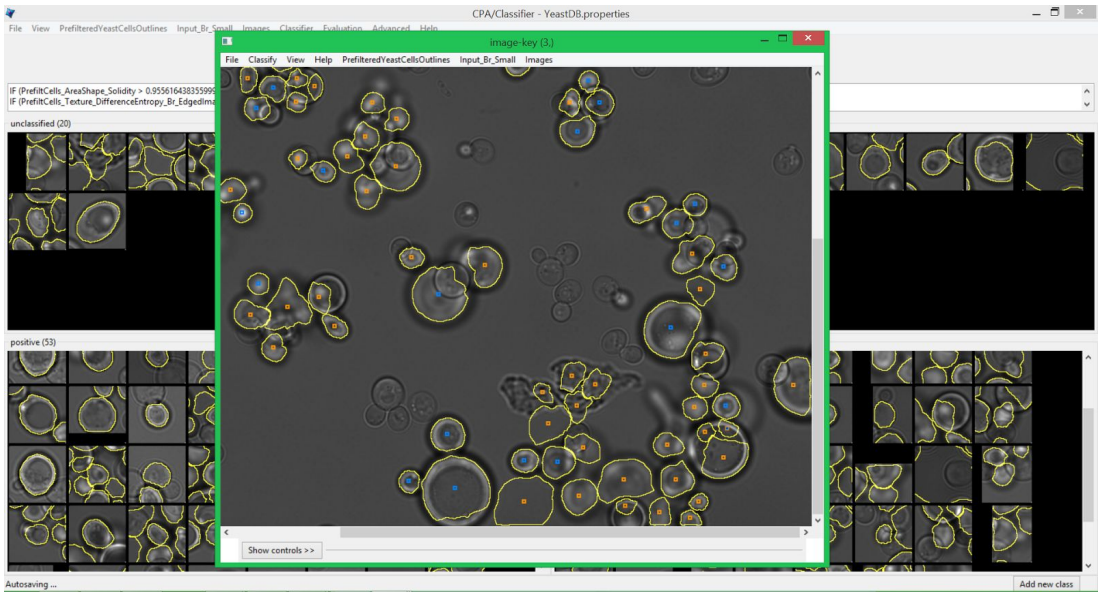


- more cases should help if 95% is better
- more rules should help if plot is growing

The look of the tiles during sorting can be changed using menu (e.g. outlines can change color or be turned off).



The complementing way to find and manually classify examples is to right click on a tile and show full image. The selected cell will be marked with a small white circle. The classifier can be run on



the entire image and errors fixed using drag-and-drop to the correct bins.

The result of the training should be rules.txt file with rules which have to manually copied from rules text box (right click and select all) and saved.

4. Save training set to save work.
5. Copy and save Rules.txt to file further processing.

Input:
- Cell outlines for guidance in CPA
- Brightfield image for guidance in CPA
- YeastDB SQL-lite database for CPA
- YeastDB.* setting files for CPA

Output:
- Rules.txt file with rules that CP can use in FilterObjects module
- TrainingSet.csv with the training sets so it can investigated, modified later

Q & A:

1. Why I cannot see any tiles or images in CPA?

Before running CPA you need to setup you YeastDB.properties file. Propably CPA cannot find/load input images. If you are using multi-page TIF files then you need to provide CPA with single-page TIF (e.g. of only brightfield)