Introduction

CellProfiler is a free Open Source software for automated image analysis. Versions for Mac, Windows and Linux are available and can be downloaded at: http://www.cellprofiler.org/. CellProfiler was developed by Anne Carpenter and Thouis Jones in the laboratory of David Sabatini at the Whitehead Institute for Biomedical Research and by Polina Golland at the CSAIL of the MIT.

CellProfiler is designed to analyze large amounts of images automatically. Since the analysis of thousands of images is very computer intensive, a version is available to run on a Linux cluster allowing the parallelization of the analysis. The latest release of CellProfiler (January 2014, version 2.1) is multicore and the number of processors used can be adjusted in the preference settings. The software is written in the Python programming language and the source code can be downloaded and extended by the community.

CellProfiler has an intuitive user interface with point and click actions that allow assembling modules in a pipeline for processing images. Several features in CellProfiler make it an easy-to-use software. The website hosts many tutorials and examples of image processing pipelines together with the corresponding images. In the GUI of CellProfiler, every button or drop down box for adjusting parameters has a help button, allowing to quickly look up its functionality.

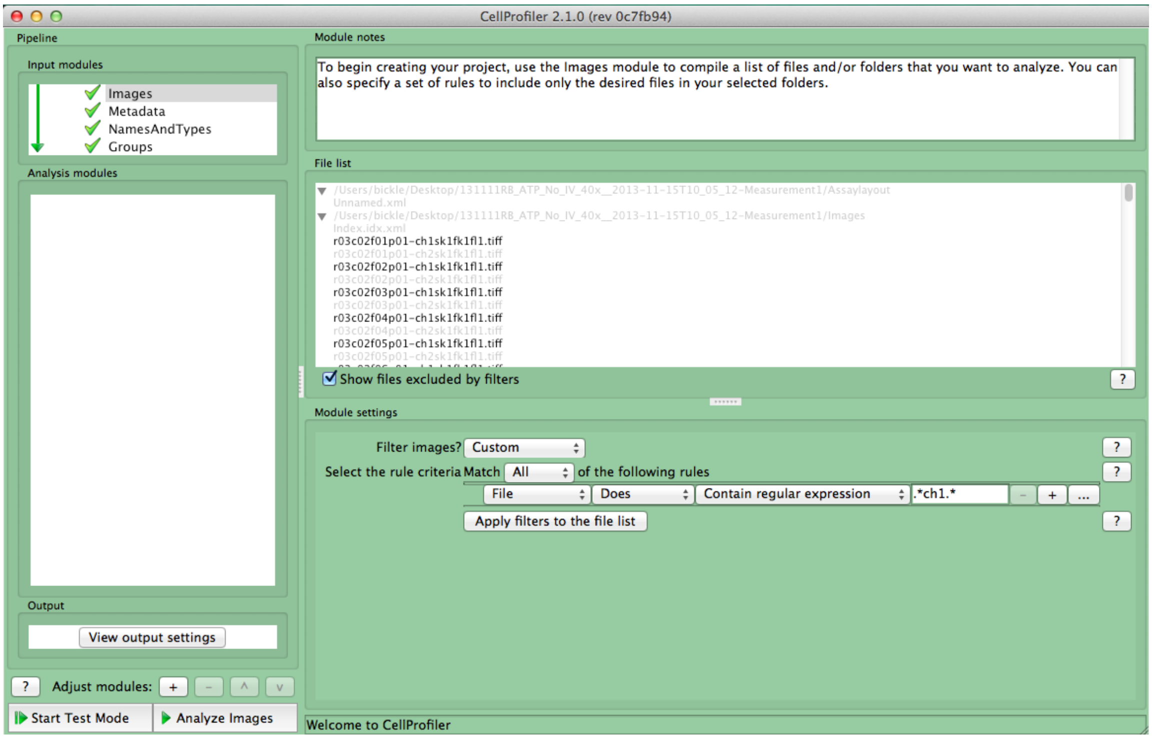
CellProfiler provides many advanced image processing algorithms allowing developing image analysis solutions for a wide variety of assays in mammalian cells, tissues slices and small organisms. There are several modules for pre-processing images such as calculating illumination correction functions and applying them to images or measuring image quality. Other modules perform image processing functions such as morphological operations (opening, closing of the image etc.), enhancement of edges and many more. A small collection of nodes is dedicated to detect objects in images using many different algorithms for distinguishing background from foreground and drawing separation lines between objects. Lastly, features such as intensity, texture and morphology can be extracted on either a object-by-object or on an image-by-image basis and exported to either CSV files, a MySQLite or MySQL database.

Visual control of the performance of the image analysis is essential in large scale screens to verify the segmentation quality of the algorithms. CellProfiler generates overlays of the detected object outlines on images, for instances on RGB color merges of the various channels created within CellProfiler or directly on the original images. Another useful feature of the software is the ability to annotate images and to extract metadata either from the path or the file name itself. The handling of the data in the statistical analysis is made much easier and allows to quickly perform quality control checks of the images and the assay.

The following document sketches a very simple application to detect nuclei. The images were acquired with the Perkin Elmer Operetta. People interested in using more powerful features or solve more difficult challenges are strongly encouraged to take advantage of the online material or take a course.

Loading Images

CellProfiler can read many image formats using the BIO-Formats library of LOCI. Images are read using the import modules found on the user interface. Alternatively the deprecated module ‘LoadImages’ can be used.
Drag the folder containing images on the field intended for this. Cellprofiler automatically searches for image files. If only a subset of images need to be analyzed, finer control over the list of images can be obtained by constructing logic filter with regular expression. In this case, only the Hoechst channel images will be loaded by selecting files with the regular expression: .*ch1.*



CellProfiler 2.1.0 (rev 0c7fb94)

Module notes

To begin creating your project, use the Images module to compile a list of files and/or folders that you want to analyze. You can also specify a set of rules to include only the desired files in your selected folders.

File list

```
▼ /Users/bickle/Desktop/131111RB_ATP_No_IV_40x__2013-11-15T10_05_12-Measurement1/Assaylayout
  Unnamed.xml
▼ /Users/bickle/Desktop/131111RB_ATP_No_IV_40x__2013-11-15T10_05_12-Measurement1/Images
  Index.idx.xml
  r03c02f01p01-ch1sk1fk1fl1.tiff
  r03c02f01p01-ch2sk1fk1fl1.tiff
  r03c02f02p01-ch1sk1fk1fl1.tiff
  r03c02f02p01-ch2sk1fk1fl1.tiff
  r03c02f03p01-ch1sk1fk1fl1.tiff
  r03c02f03p01-ch2sk1fk1fl1.tiff
  r03c02f04p01-ch1sk1fk1fl1.tiff
  r03c02f04p01-ch2sk1fk1fl1.tiff
  r03c02f05p01-ch1sk1fk1fl1.tiff
  r03c02f05p01-ch2sk1fk1fl1.tiff
```

☑ Show files excluded by filters

Module settings

Filter images? Custom

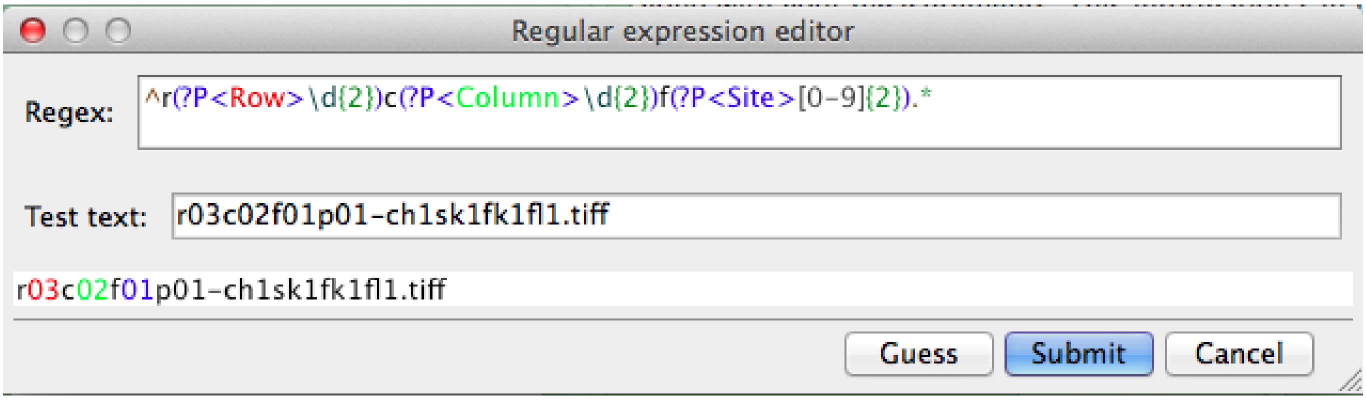
Select the rule criteria Match All of the following rules

File | Does | Contain regular expression | .*ch1.*

Apply filters to the file list

Welcome to CellProfiler

Next metadata contained in the path to the images and their name is defined and extracted using again regular expressions.



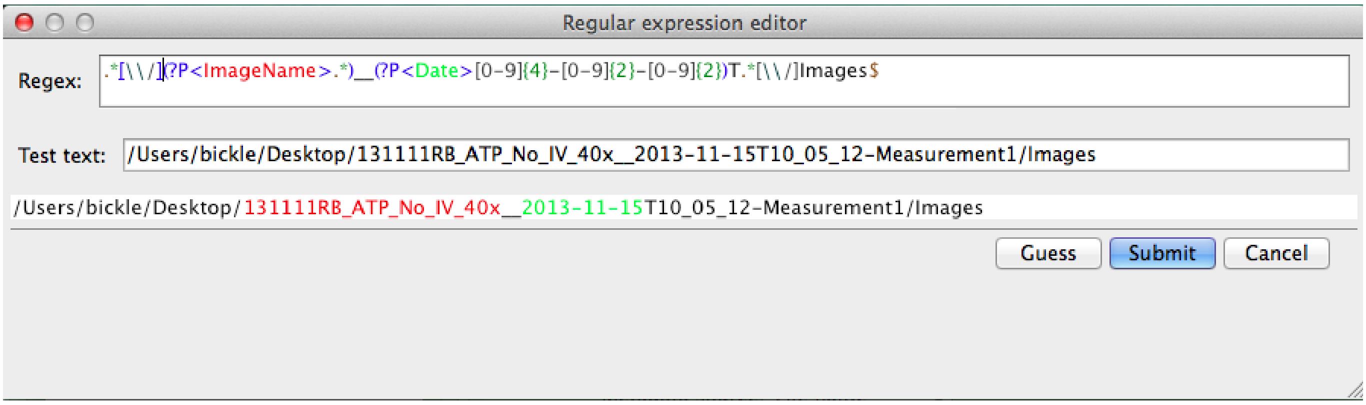
Regular expression editor

Regex: ^r(?P<Row>\d{2})c(?P<Column>\d{2})f(?P<Site>[0-9]{2}).*

Test text: r03c02f01p01-ch1sk1fk1fl1.tiff

r03c02f01p01-ch1sk1fk1fl1.tiff

Metadata extraction from file name



Regular expression editor

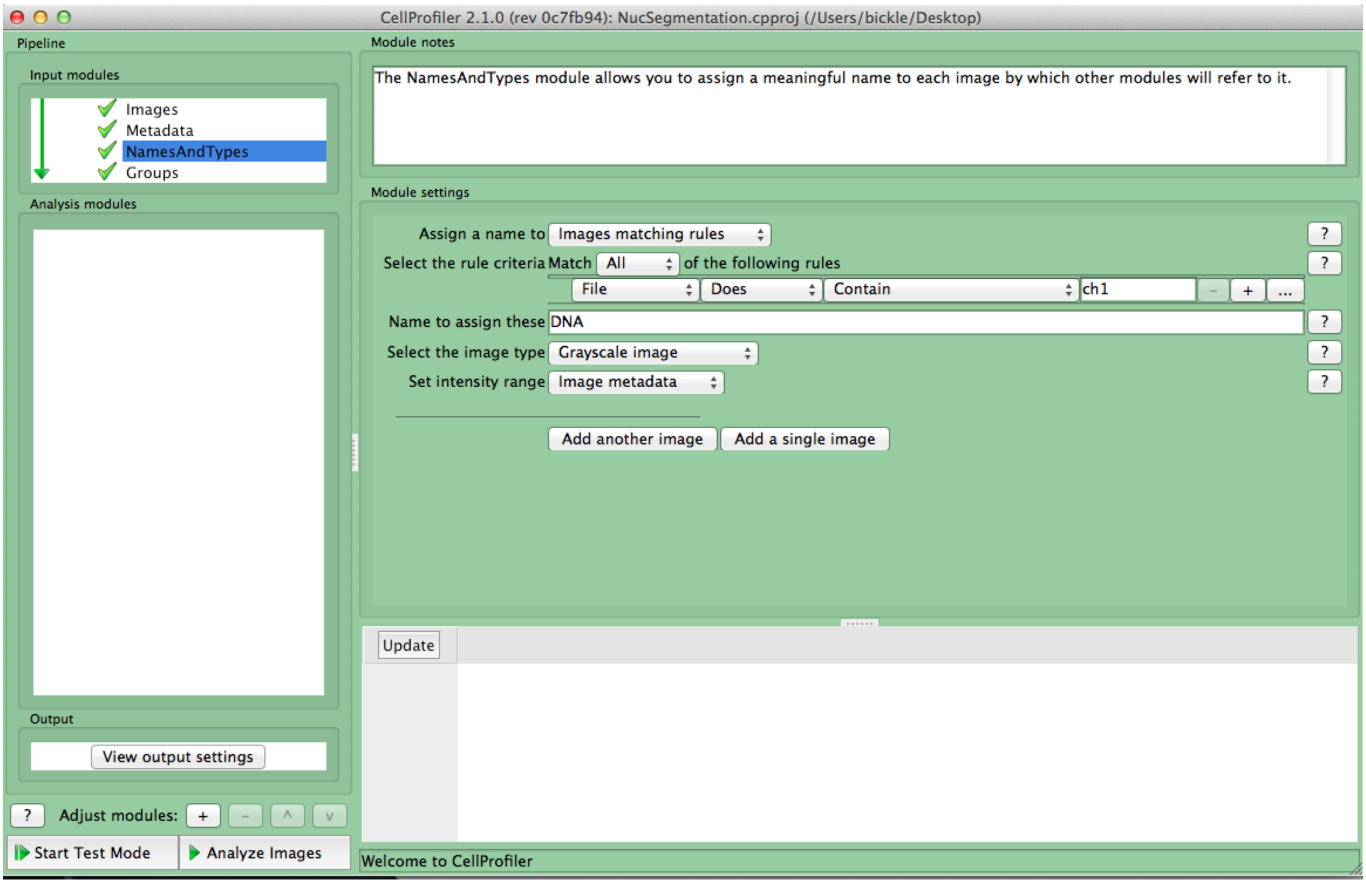
Regex: `.*[\\/](?P<ImageName>.*)__(?P<Date>[0-9]{4}-[0-9]{2}-[0-9]{2})T.*[\\/]Images$`

Test text: /Users/bickle/Desktop/131111RB_ATP_No_IV_40x__2013-11-15T10_05_12-Measurement1/Images

/Users/bickle/Desktop/131111RB_ATP_No_IV_40x__2013-11-15T10_05_12-Measurement1/Images

Metadata extraction from image path

A name variable has then to be associated with the loaded images.



CellProfiler 2.1.0 (rev 0c7fb94): NucSegmentation.cpproj (/Users/bickle/Desktop)

Module notes: The NamesAndTypes module allows you to assign a meaningful name to each image by which other modules will refer to it.

Module settings:

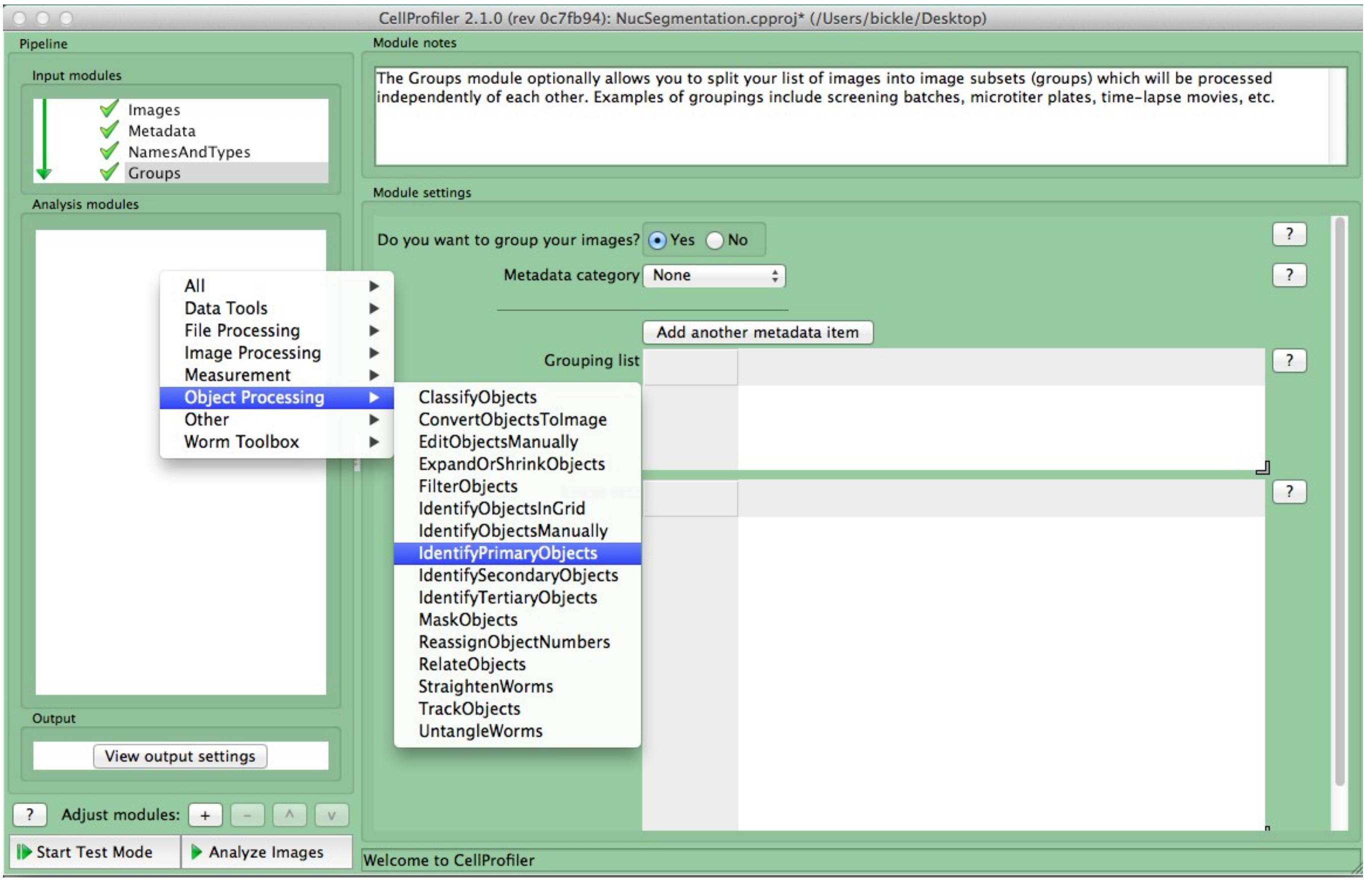
- Assign a name to: Images matching rules
- Select the rule criteria: Match All of the following rules
- File | Does | Contain | ch1
- Name to assign these: DNA
- Select the image type: Grayscale image
- Set intensity range: Image metadata

Welcome to CellProfiler

Lastly, the association of the images for multi tif stacks and movies can be specified, but is not applicable here.

Segmenting objects

The most crucial and computer intensive step in image analysis is object segmentation. A common task in most image analysis routines is to detect nuclei as primary objects and cytoplasm of cells as secondary objects around the nuclei. We generally stain nuclei with Hoechst and the cytoplasm with HCS CellMask Blue (Life Technologies). Nuclei are detected using Otsu's three class thresholding and fused objects are separated by finding local maxima followed by watershed. These methods are available in the 'IdentifyPrimaryObjects' module.



CellProfiler 2.1.0 (rev 0c7fb94): NucSegmentation.cpproj* (/Users/bickle/Desktop)

Module notes

The Groups module optionally allows you to split your list of images into image subsets (groups) which will be processed independently of each other. Examples of groupings include screening batches, microtiter plates, time-lapse movies, etc.

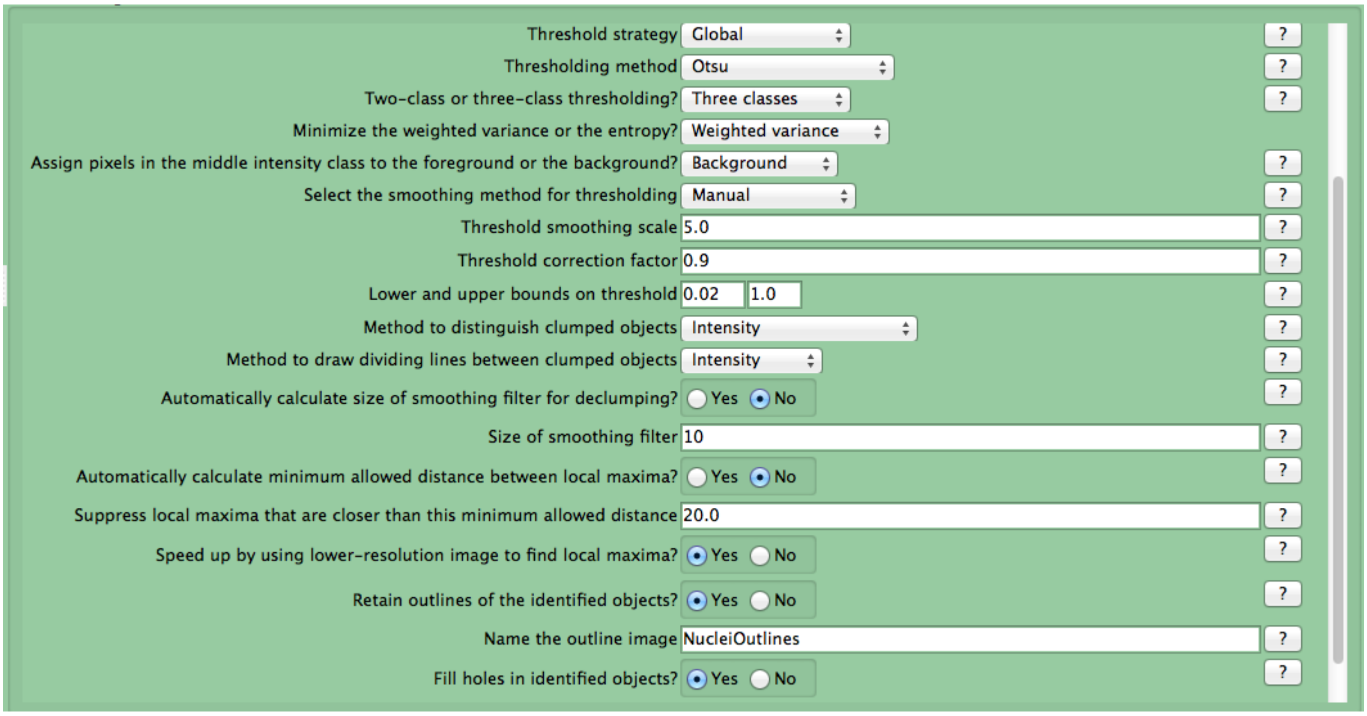
Module settings

Do you want to group your images? ● Yes ○ No

Metadata category: None

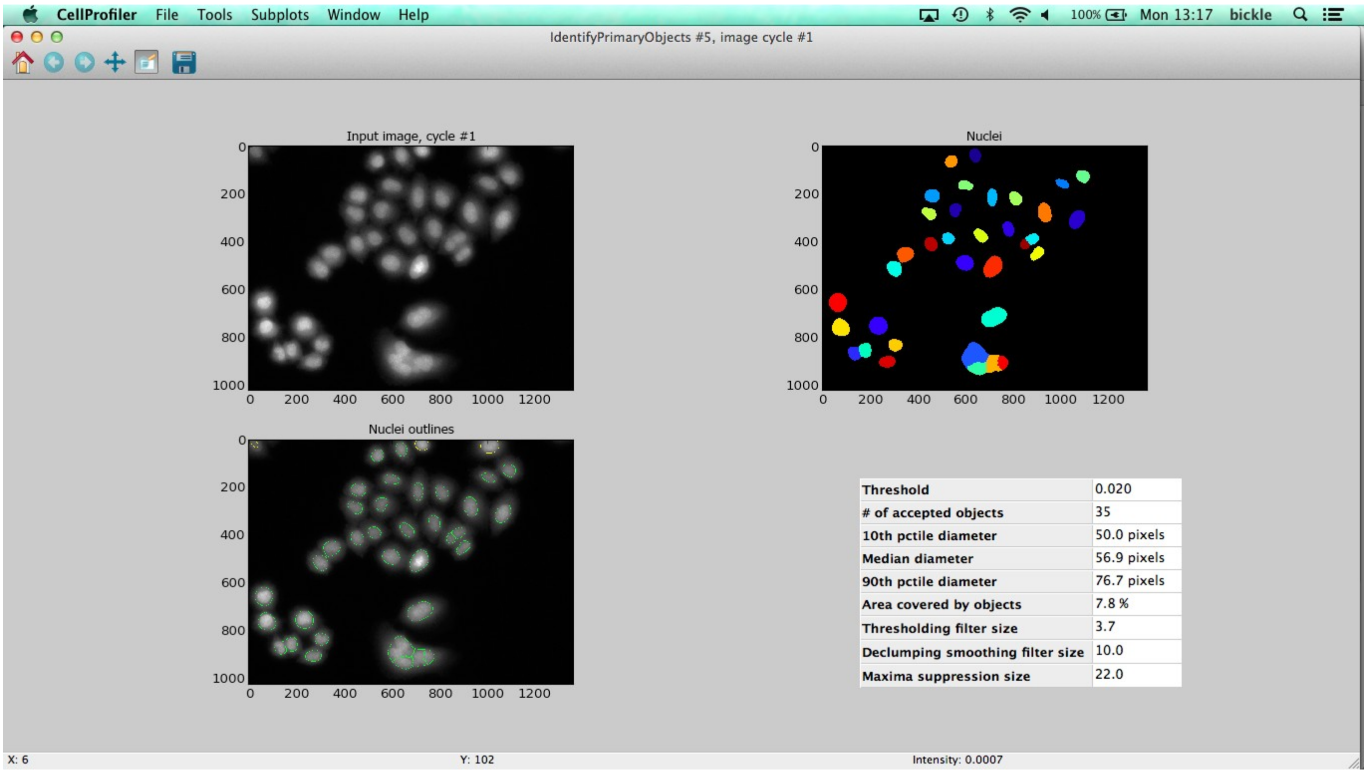
Welcome to CellProfiler

The module is adjusted from top to bottom selecting the image to analyze, the name of the output objects, their size etc. For each customization step, help is available by clicking on the corresponding question mark. CellProfiler offers the option to automatically find the best segmentation method. While this is generally very efficient, we typically do not use it, since we wish to apply the same segmentation algorithm to all of our images and do not want signal variation due to different algorithms or settings. We typically use Otsu three level segmentation, set a lower threshold to define the segmentation levels and define smoothing filters and minimal distance between local maxima. We factorize all parameters to avoid different settings between images and to maintain consistency across batches. We also save the outlines of the identified objects in order to control for the correctness of the segmentation.



Threshold strategy	Global
Thresholding method	Otsu
Two-class or three-class thresholding?	Three classes
Minimize the weighted variance or the entropy?	Weighted variance
Assign pixels in the middle intensity class to the foreground or the background?	Background
Select the smoothing method for thresholding	Manual
Threshold smoothing scale	5.0
Threshold correction factor	0.9
Lower and upper bounds on threshold	0.02 1.0
Method to distinguish clumped objects	Intensity
Method to draw dividing lines between clumped objects	Intensity
Automatically calculate size of smoothing filter for declumping?	○ Yes ● No
Size of smoothing filter	10
Automatically calculate minimum allowed distance between local maxima?	○ Yes ● No
Suppress local maxima that are closer than this minimum allowed distance	20.0
Speed up by using lower-resolution image to find local maxima?	● Yes ○ No
Retain outlines of the identified objects?	● Yes ○ No
Name the outline image	NucleiOutlines
Fill holes in identified objects?	● Yes ○ No

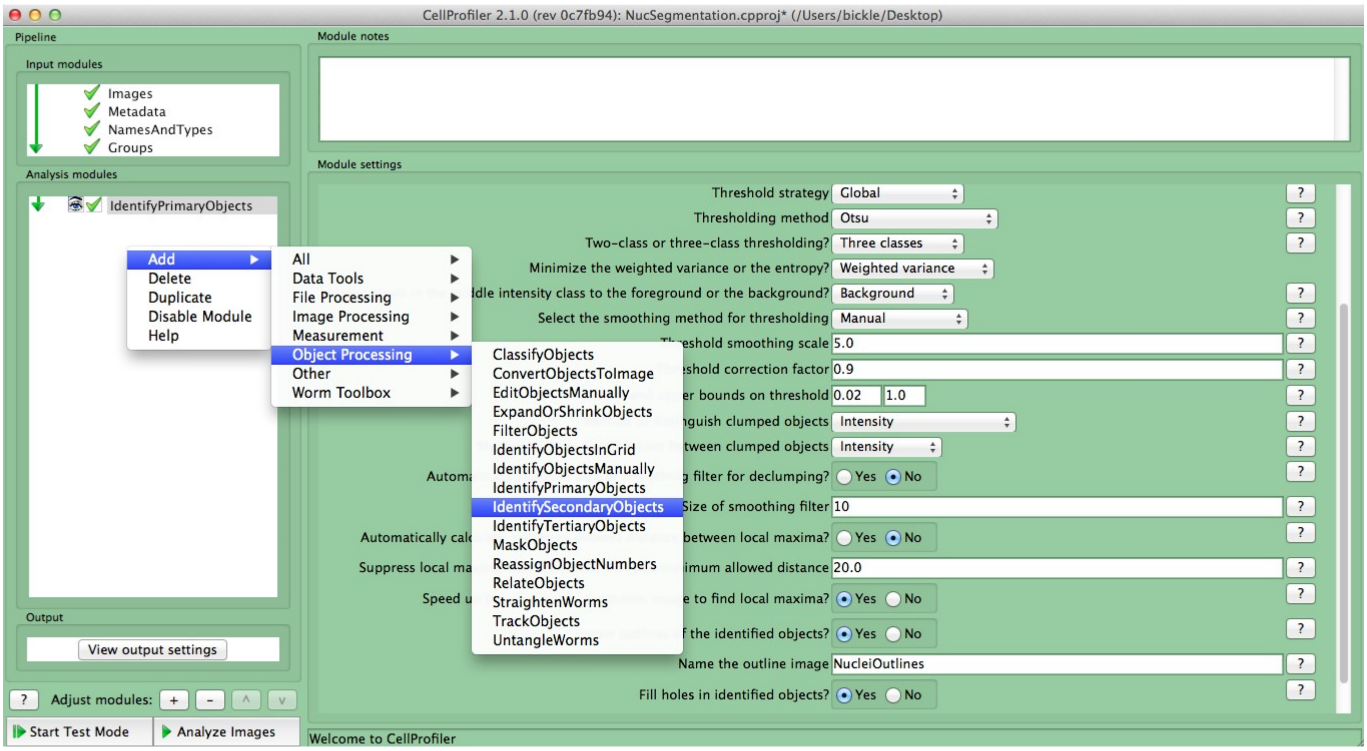
Result of the segmentation looks like this:


CellProfiler File Tools Subplots Window Help
100% Mon 13:17 bickle
IdentifyPrimaryObjects #5, image cycle #1
Input image, cycle #1
Nuclei
Nuclei outlines
0
200
400
600
800
1000
1200

Threshold	0.020
# of accepted objects	35
10th pctile diameter	50.0 pixels
Median diameter	56.9 pixels
90th pctile diameter	76.7 pixels
Area covered by objects	7.8 %
Thresholding filter size	3.7
Declumping smoothing filter size	10.0
Maxima suppression size	22.0

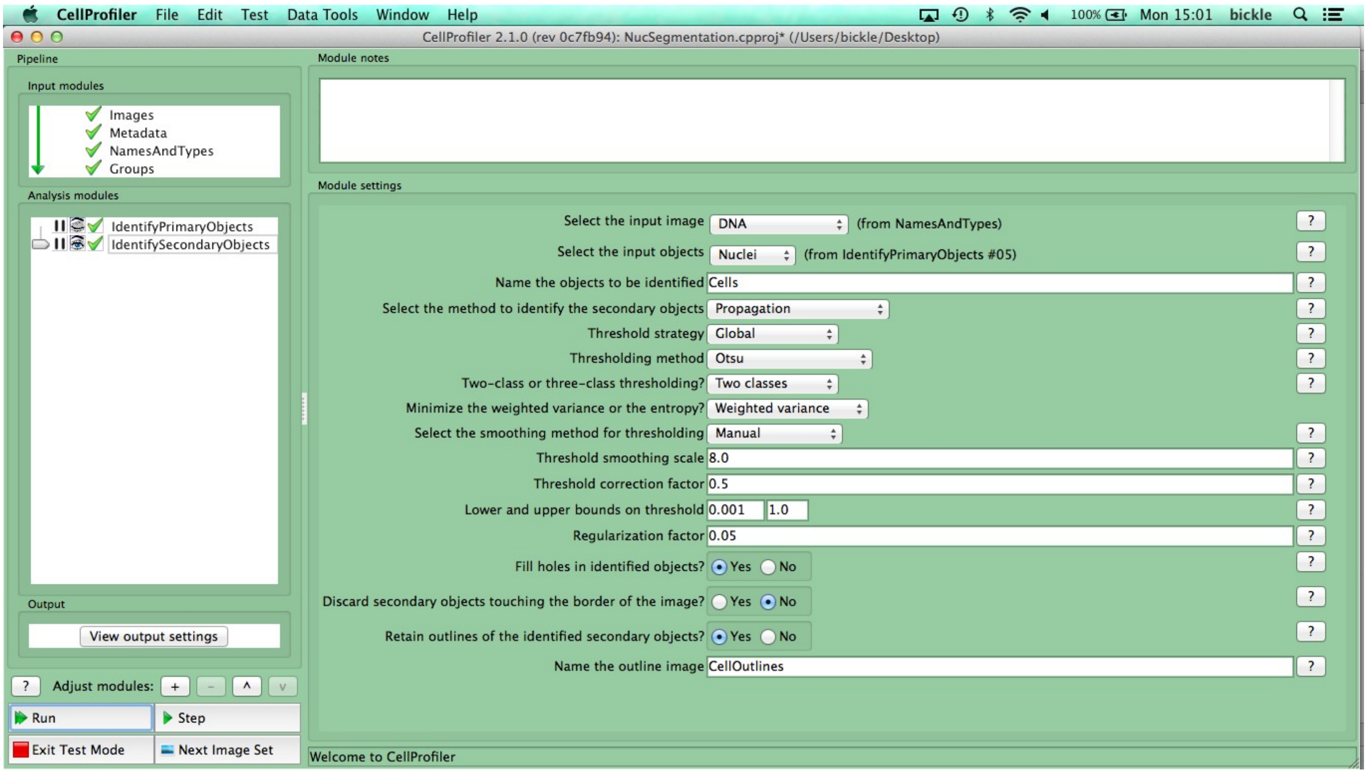
X: 6 Y: 102 Intensity: 0.0007

Next, using the ‘DetectSecondaryObjects’ module, the stained cytoplasms are detected.



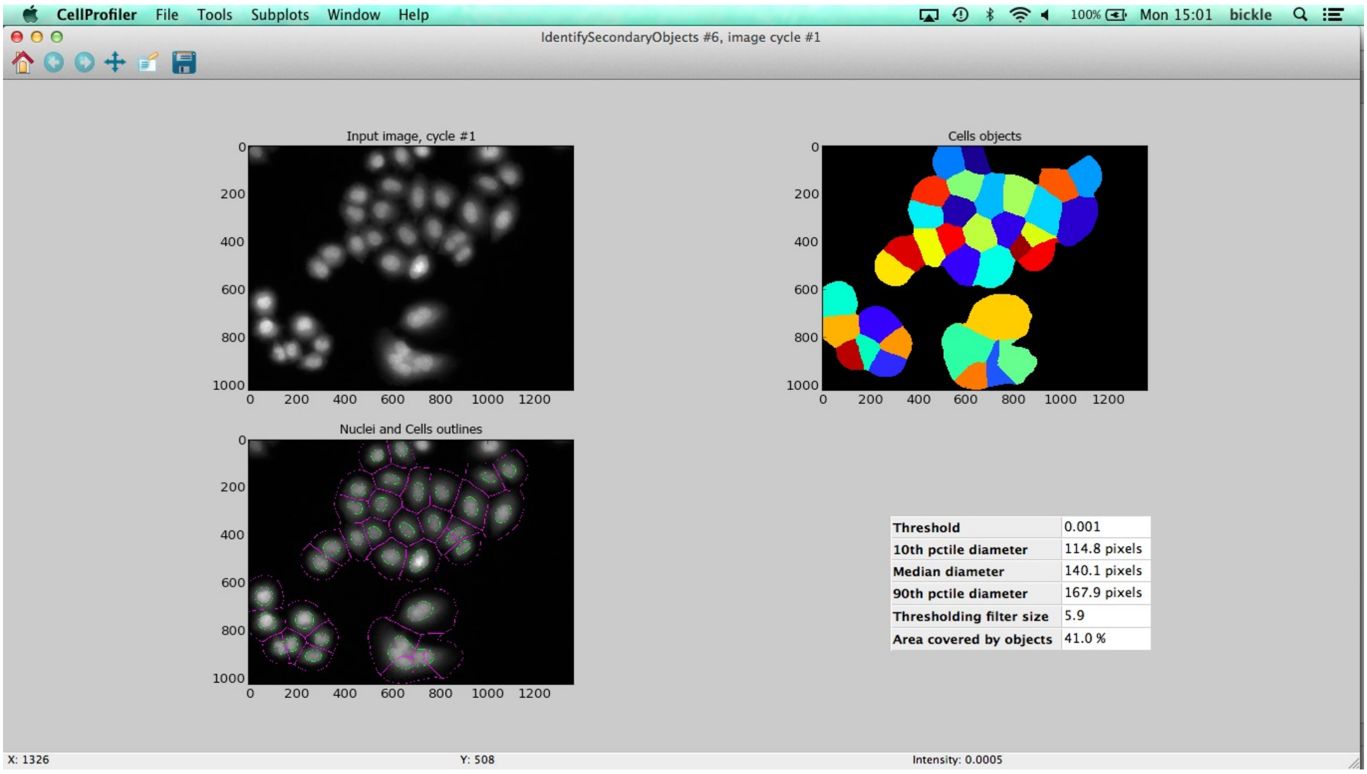
Threshold strategy: Global
Thresholding method: Otsu
Two-class or three-class thresholding? Three classes
Minimize the weighted variance or the entropy? Weighted variance
…ddle intensity class to the foreground or the background? Background
Select the smoothing method for thresholding: Manual
…eshold smoothing scale: 5.0
…shold correction factor: 0.9
…er bounds on threshold: 0.02 1.0
…guish clumped objects: Intensity
…tween clumped objects: Intensity
Autom… …g filter for declumping? ○ Yes ● No
…Size of smoothing filter: 10
Automatically cal… …between local maxima? ○ Yes ● No
Suppress local ma… …imum allowed distance: 20.0
Speed u… …e to find local maxima? ● Yes ○ No
…f the identified objects? ● Yes ○ No
Name the outline image: NucleiOutlines
Fill holes in identified objects? ● Yes ○ No

We typically use the propagation algorithm and the nuclei as seeds. The local thresholds are again determined by using Otsu and all parameters are determined.



Select the input image: DNA (from NamesAndTypes)
Select the input objects: Nuclei (from IdentifyPrimaryObjects #05)
Name the objects to be identified: Cells
Select the method to identify the secondary objects: Propagation
Threshold strategy: Global
Thresholding method: Otsu
Two-class or three-class thresholding? Two classes
Minimize the weighted variance or the entropy? Weighted variance
Select the smoothing method for thresholding: Manual
Threshold smoothing scale: 8.0
Threshold correction factor: 0.5
Lower and upper bounds on threshold: 0.001 1.0
Regularization factor: 0.05
Fill holes in identified objects? ● Yes ○ No
Discard secondary objects touching the border of the image? ○ Yes ● No
Retain outlines of the identified secondary objects? ● Yes ○ No
Name the outline image: CellOutlines

The result is the following:

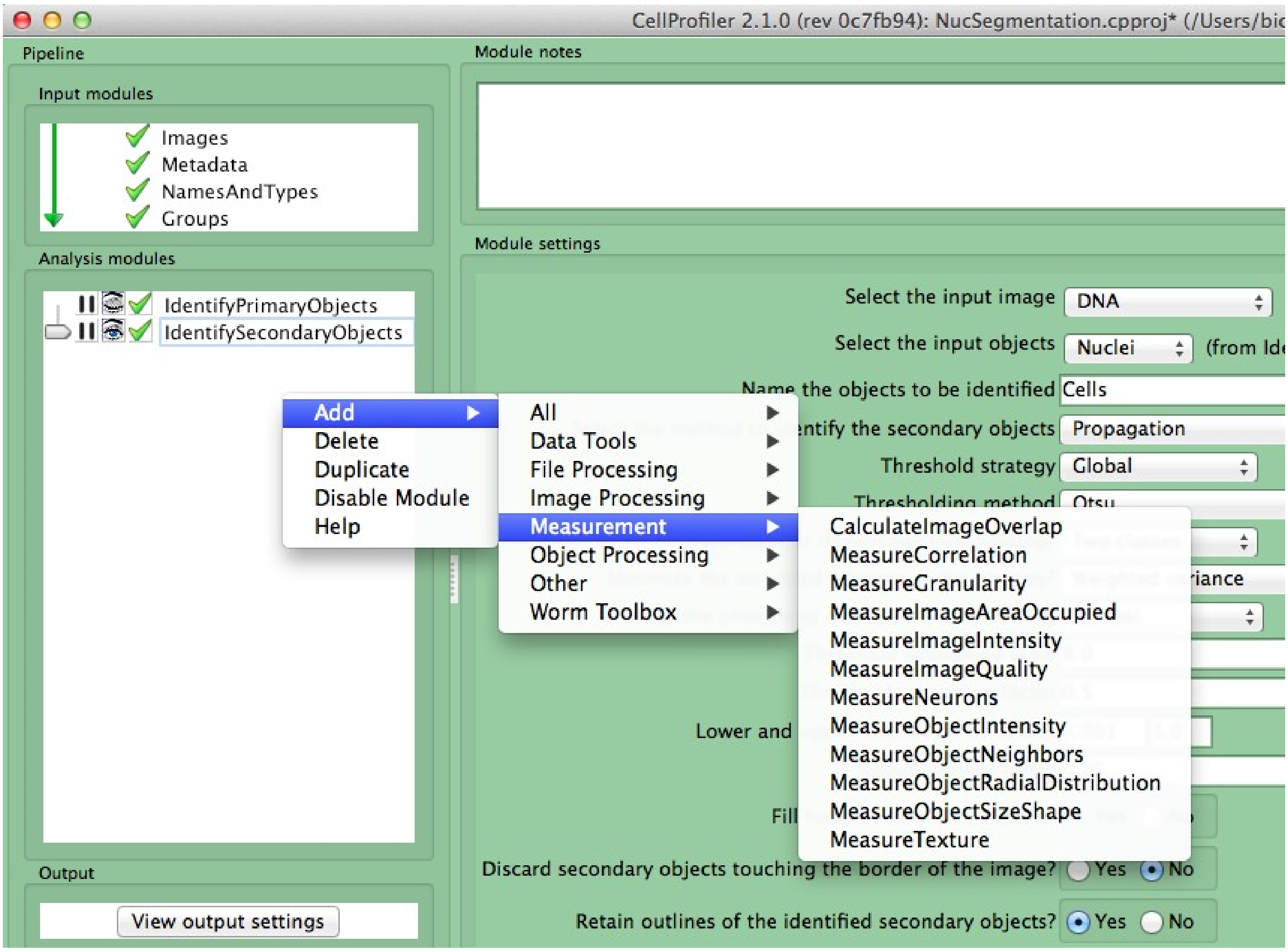

CellProfiler File Tools Subplots Window Help
100% Mon 15:01 bickle
IdentifySecondaryObjects #6, image cycle #1
Input image, cycle #1
Cells objects
Nuclei and Cells outlines
0
200
400
600
800
1000
1200
X: 1326
Y: 508
Intensity: 0.0005

Threshold	0.001
10th pctile diameter	114.8 pixels
Median diameter	140.1 pixels
90th pctile diameter	167.9 pixels
Thresholding filter size	5.9
Area covered by objects	41.0 %

A further common task in many biological assays is to detect smaller objects such as organelles in the cytoplasm of cells. For small structures with high background intensities, a white top-hat filter is applied using the ‘EnhanceOrSuppressFeatures’ module to increase the contrast. The tertiary objects are then detected, using the ‘IdentifyPrimaryObject’ module with multi threshold Otsu and watershed seeded from local maxima. Finally, to obtain tertiary object data on a cell-by-cell basis, the objects are linked to their parent cell with a ‘RelateObjects’ module.

Extracting parameters

After segmenting all the objects of interest, features are extracted for the quantitative phenotypic description. Several measurement modules are available in CellProfiler for both image-based measurements and object-based measurements.



Select the input image DNA

Select the input objects Nuclei (from Ide

Name the objects to be identified Cells

...ntify the secondary objects Propagation

Threshold strategy Global

Thresholding method Otsu

Lower and

Fill

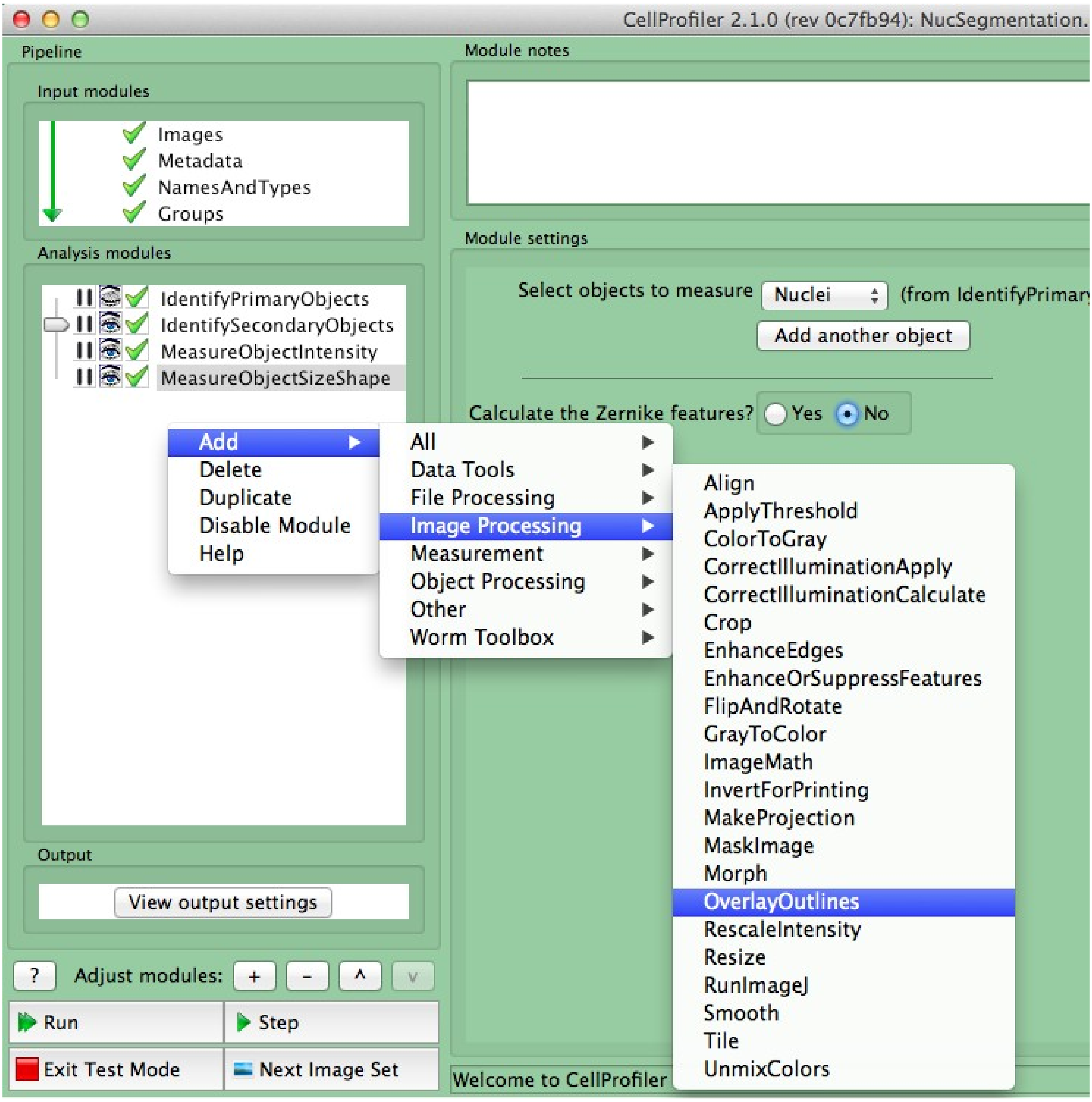
Discard secondary objects touching the border of the image? ○ Yes ● No

Retain outlines of the identified secondary objects? ● Yes ○ No

Using a ‘MeasureObjectIntensity’ module, many aspects of the intensity of objects can be captured such as, their integral, average or median intensity. Other modules measure the morphology (MeasureObjectSizeShape), the texture (MeasureTexture) or the spatial distribution of objects (MeasureObjectRadialDistribution, MeasureObjectNeighbors). Extracting all these parameters can easily generate tables of several hundreds of columns.

Control images and exporting data

When analyzing large screens, it is impossible to visually control the quality of segmentation of all the images. A spot control needs to be carried out and control images showing the outlines of the segmented objects are very useful for this. Furthermore, it is paramount to visualize all the images of hit wells to ensure that no spurious segmentation artifact leads to the obtained result. To create control images the ‘OverlayOutlines’ module is added.



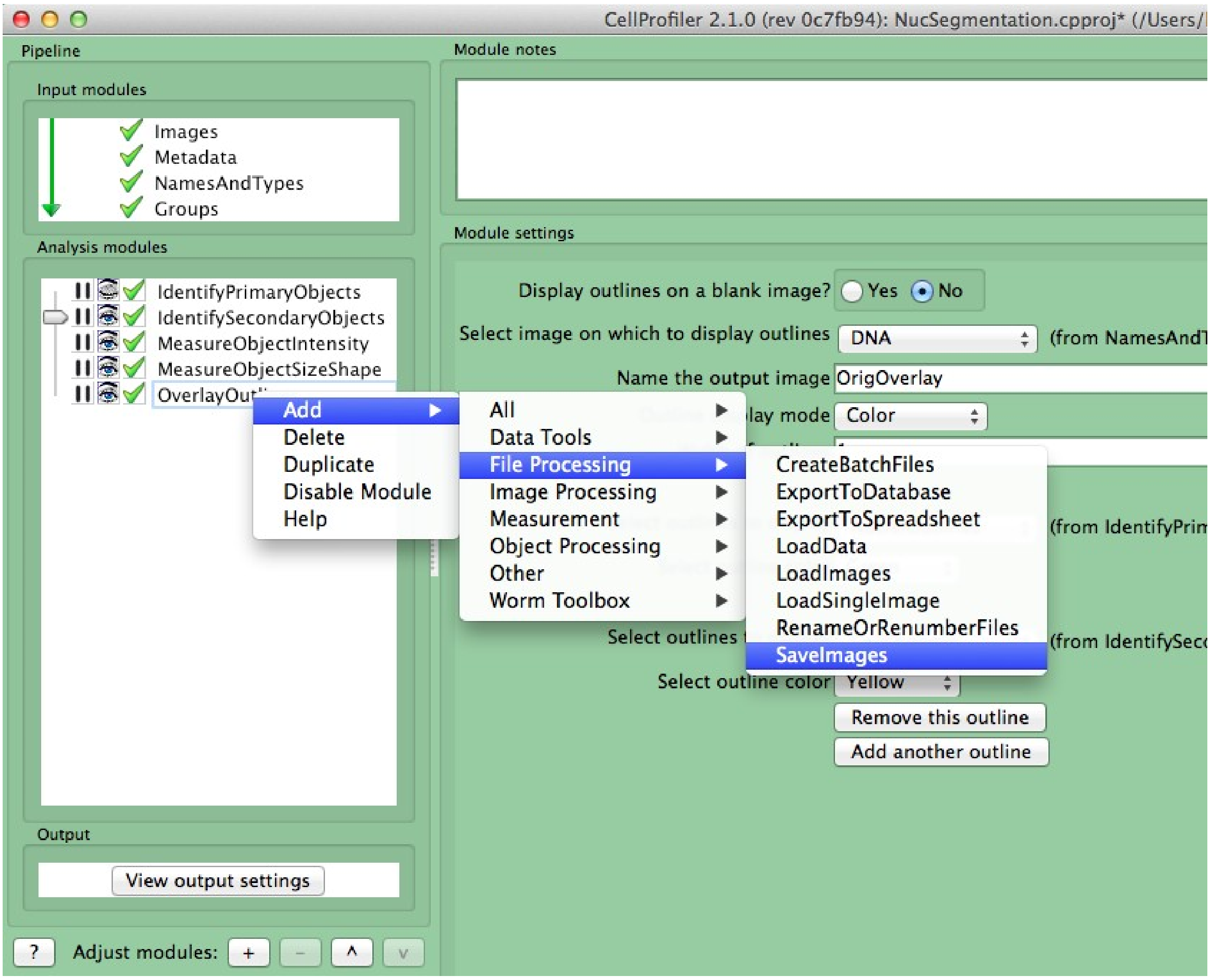
CellProfiler 2.1.0 (rev 0c7fb94): NucSegmentation.

Select objects to measure: Nuclei (from IdentifyPrimar

Calculate the Zernike features? ○ Yes ● No

Welcome to CellProfiler

The outlines of each segmented object can be overlaid on an image and saved with the ‘SaveImages’ module.



Display outlines on a blank image? ○ Yes ● No

Select image on which to display outlines: DNA (from NamesAndT

Name the output image: OrigOverlay

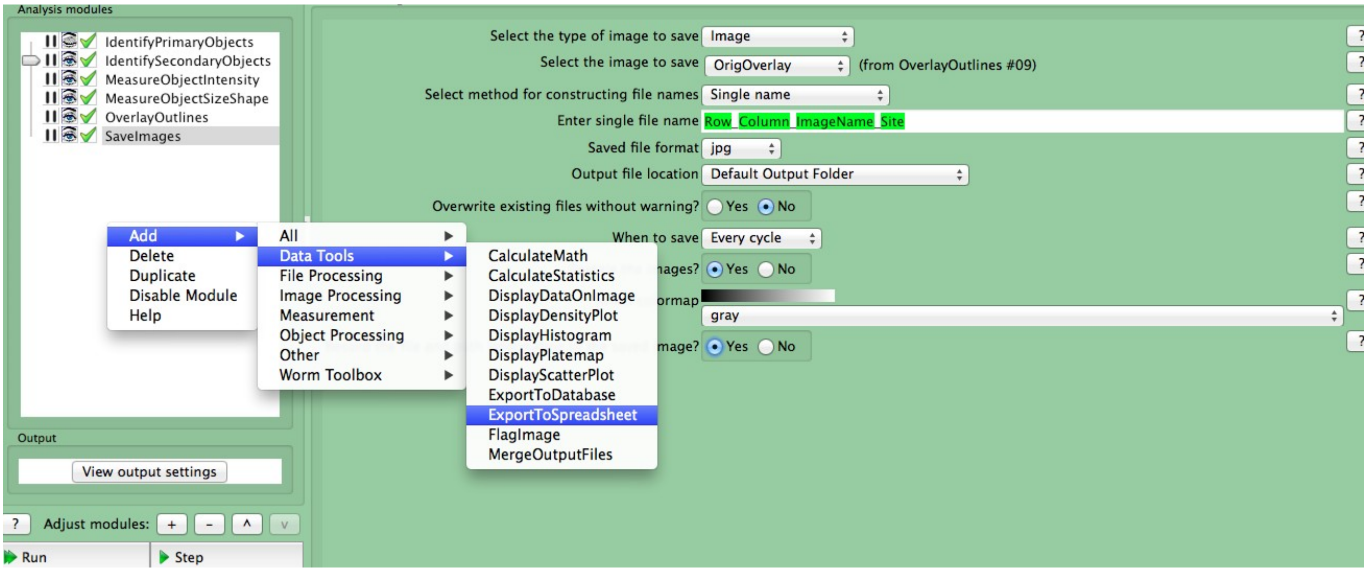
lay mode: Color

(from IdentifyPrim

Select outlines (from IdentifySec

Select outline color: Yellow

The last step of the analysis exports the data and associated metadata as comma-separated values file (CSV files) using the ‘ExportToSpreadsheet’ module or to a database using the ‘ExportToDatabase’ module.



Select the type of image to save: Image

Select the image to save: OrigOverlay (from OverlayOutlines #09)

Select method for constructing file names: Single name

Enter single file name: Row_Column_ImageName_Site

Saved file format: jpg

Output file location: Default Output Folder

Overwrite existing files without warning? ○ Yes ● No

When to save: Every cycle

ages? ● Yes ○ No

ormap: gray

mage? ● Yes ○ No

Conclusion

This simple pipeline will allow segmenting nuclei and extract parameters about the segmented objects. For more complex pipelines, the tutorial on the CellProfiler website should be followed. Alternatively, the TDS of the Max Planck Institute of Molecular Cell Biology and Genetics in Dresden offers one day courses on CellProfiler (please contact Marc Bickle: bickle@mpi-cbg.de).