

BLEND - BLEbbed Nuclei Detector

FIJI / IMAGEJ PLUGIN

**Accurate Detection of Dymorphic Nuclei Using
Dynamic Programming and Supervised Classification**

Marlies Verschuuren ^{1,2}, Jonas De Vylder ³, Hannes Catrysse ⁴, Wilfried Philips ³, Winnok H. De Vos ^{1,5*}

¹ Department of Veterinary Sciences, University of Antwerp, Belgium

² Ph. D. fellowship of the Research Foundation - Flanders (FWO)

³ Department of Telecommunication and Information Processing, IPI, iMinds, Ghent University, Belgium

⁴ Institute for Agricultural and Fisheries Research (ILVO), Melle, Belgium

⁵ Department of Molecular Biotechnology, Ghent University, Belgium

*winnok.devos@uantwerpen.be

ABOUT

Cancer, aging and viral infection, have one major common denominator at the cellular level: nuclear dysmorphism. Next to overt modifications in size and overall shape, subtle protrusions, called blebs, are often indicative of a local weakening of the nucleus and represent a strong predictor of pathology development. Accurate detection of dysmorphic nuclei may therefore be a valuable tool in routine screening and intelligent imaging approaches. However, due to lack of prior knowledge about their size, shape and intensity, dysmorphic nuclei are often not accurately detected in standard image analysis routines. To enable accurate detection of dysmorphic nuclei, we have implemented an automated image analysis algorithm, called BLEND (BLEbbed Nuclei Detector), which is based on two-tier seed detection and dynamic programming.

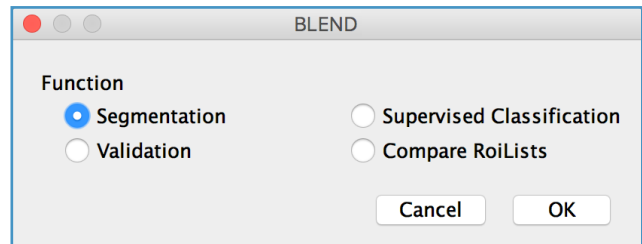
INSTALL

BLEND is implemented as a java plugin in the software platform FIJI. Fiji can be downloaded from www.fiji.sc.

BLEND can be downloaded from: <https://github.com/VerschuurenM/BLEND>

Drag 'n drop the “https://github.com/VerschuurenM/BLEND/blob/master/target/BLEND_Plugin-0.2.1-SNAPSHOT.jar” file into the FIJI user interface and save. After restarting FIJI, BLEND can be called from **Plugins > BLEND**

FUNCTIONS



The following functions are implemented in the plugin:

- > **Segmentation**: Segmentation of input images with selected parameter settings.
- > **Supervised Classifications**: Iterative learning process for classification of nuclei.
- > **Validation**: Validation by comparing automated segmentations with user defined ground truths.
- > **Compare roiLists**: Compare roiLists with user defined ground truth roiLists.

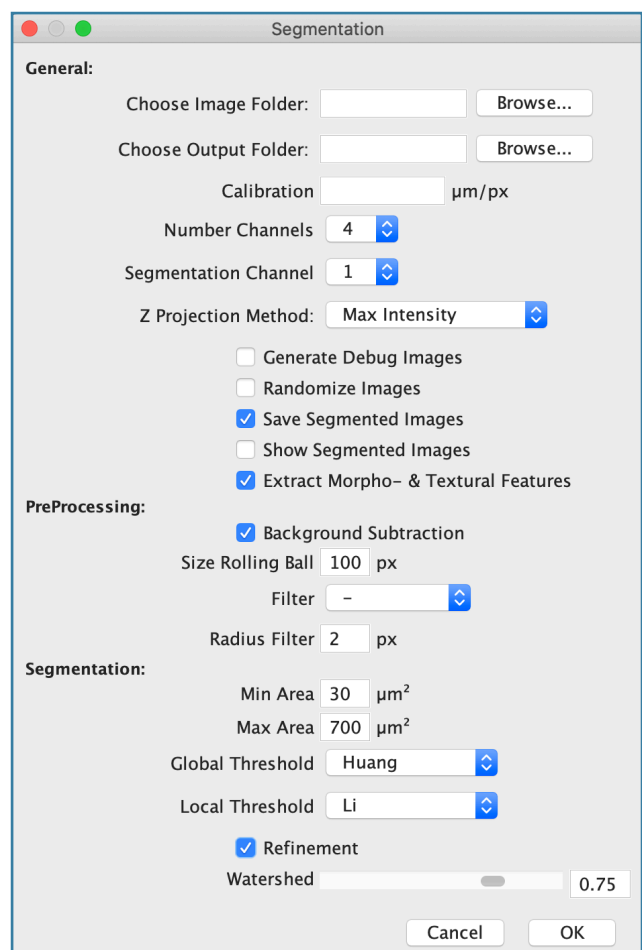
Segmentation

In order to start BLEND, an **image and output folder** have to be assigned. The user has to define the **calibration and number of channels** of the images. Furthermore the **channel used for nuclei segmentation as well as the z projection method to use** have to be selected. The user can also choose to show **debug images, randomise images, show and save segmented images** and **extract features from the detected nuclei in all channels**.

Background subtraction and multiple standard available linear and non-linear image **filters** (Gaussian, Median, Mean, Minimum, Maximum and Variance) are implemented in BLEND. The **scale** has to be defined by the user.

Segmentation results can be optimised by adjusting certain parameters. A **minimum and maximum nuclear area** has to be defined as well as a **global and local threshold** method for the 2 pass thresholding algorithm. When the local threshold method is set to '/', a 1 pass thresholding will be implemented. In addition, the user can opt to

exclude **contour refinement** from the algorithm. A final parameter that has to be set is the **watershed cutoff**. A higher cutoff will lead to more nuclei that are split.



BLEND - BLEBBED NUCLEI DETECTOR

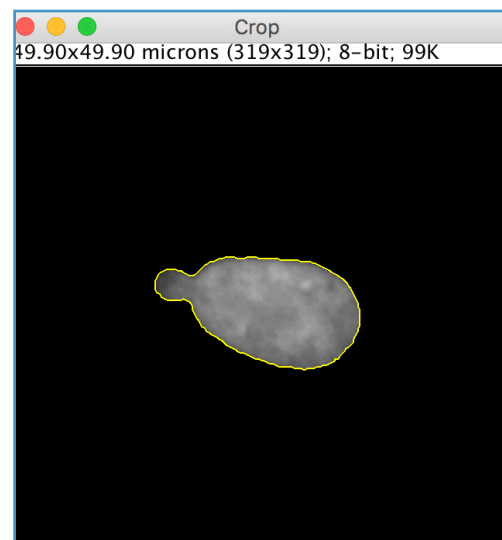
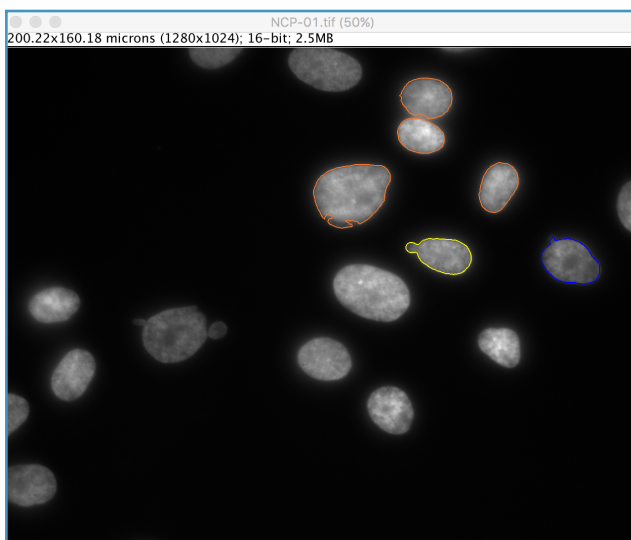
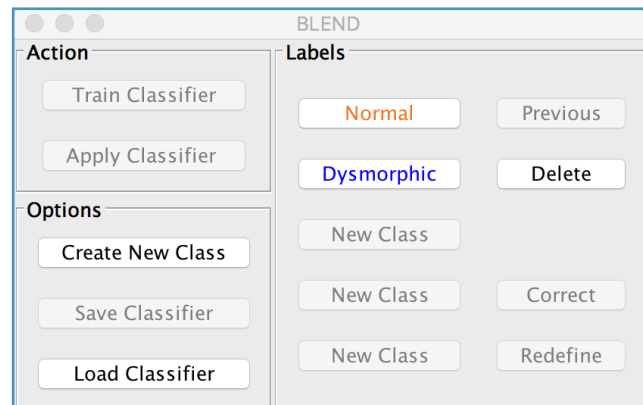
Supervised Classification

In this module, the segmentation is combined with a classification routine. An user interface allows the user to iteratively train a '*random forest*' or '*support vector machine*' classifier to discriminate nuclei in up to five different classes.

New classes can be added by clicking "**Create New Class**". The segmented nuclei can then be classified by clicking on the **label** buttons.

When a sufficient amount of nuclei is labeled, a classifier can be trained by clicking on "**Train Classifier**". The classes of the subsequent nuclei are then predicted using the trained classifier and can be assigned as **Correct** or can be **Redefined**.

The classifier can always be re-trained and saved as .model file (WEKA) (**Save Classifier**). When the results of the iterative training are satisfying, the classifier can be applied to the whole dataset by clicking "**Apply Classifier**". A previously trained classifier can also be loaded as .model file (WEKA) by clicking the "**Load Classifier**" button.

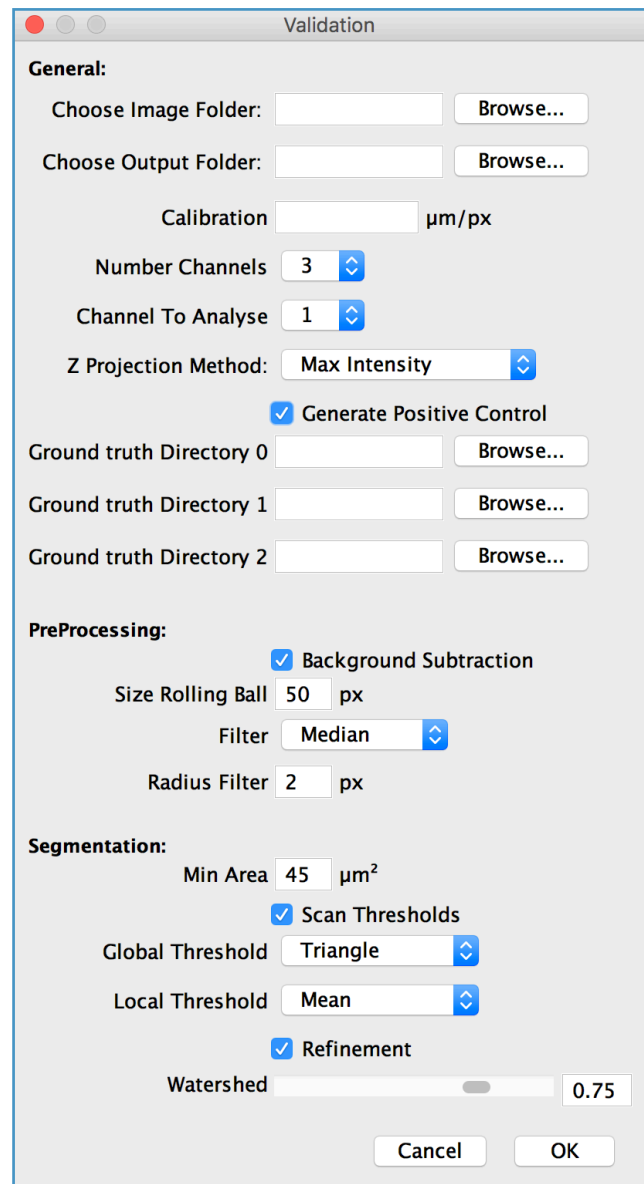


Validation

To validate the segmentation process and to identify the best threshold combinations, a module is implemented in which segmentation results are compared with manual delineated ground truths.

In addition to the previously mentioned general settings, at least one (max 3) **ground truth directory** has to be defined. In this directory the manual delineations have to be stored as "ImageX_roiSet.zip" file for each image in the image folder. These zip files can be generated with the Roi Manager implemented in FIJI. When multiple ground truth folders are chosen, a **positive control** can be generated by comparing these ground truths.

The user can choose to **scan all threshold** combinations for 2 pass and 1 pass (local = "/") thresholding or just validate 1 combination. Other settings are explained in previous sections



The Validation dialog box is organized into several sections: General, PreProcessing, and Segmentation. The General section includes fields for image and output folders, calibration, number of channels, channel to analyze, Z projection method, and ground truth directories. The PreProcessing section includes checkboxes for background subtraction and generate positive control, along with settings for rolling ball size, filter, and radius. The Segmentation section includes checkboxes for scan thresholds and refinement, and a watershed slider.

Validation

General:

Choose Image Folder: **Browse...**

Choose Output Folder: **Browse...**

Calibration $\mu\text{m}/\text{px}$

Number Channels **▾**

Channel To Analyse **▾**

Z Projection Method: **▾**

☒ **Generate Positive Control**

Ground truth Directory 0 **Browse...**

Ground truth Directory 1 **Browse...**

Ground truth Directory 2 **Browse...**

PreProcessing:

☒ **Background Subtraction**

Size Rolling Ball **px**

Filter **▾**

Radius Filter **px**

Segmentation:

Min Area μm^2

☒ **Scan Thresholds**

Global Threshold **▾**

Local Threshold **▾**

☒ **Refinement**

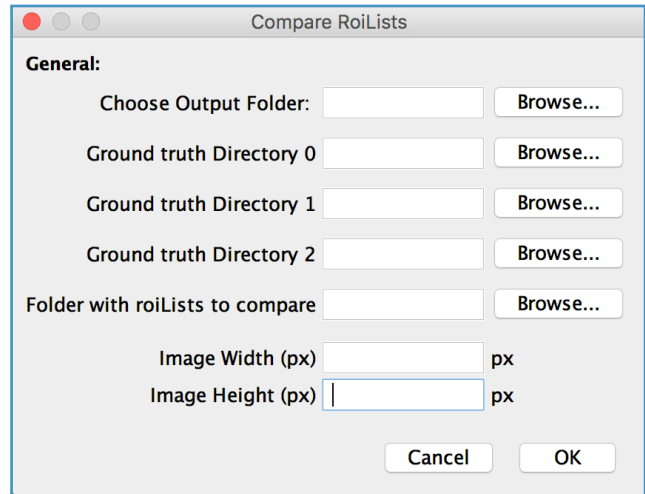
Watershed

Cancel **OK**

Compare roiLists

The validation process can also be performed with previously generated roiLists. Again an **output folder** has to be chosen as well as a **ground truth directory** (max 3). In addition to this, the user has to define the **folder in which the different roiLists are stored** as “ImageX_roiSet.zip” file.

In order to start the comparison process the image width and height have to be defined.



Compare RoiLists

General:

Choose Output Folder:

Ground truth Directory 0

Ground truth Directory 1

Ground truth Directory 2

Folder with roiLists to compare

Image Width (px) px

Image Height (px) px