

MPI-Ar-GE

Authors: Matthias Arzt, Joran Deschamps, Florian Jug

Email: artz@mpi-cbg.de

Platform: Linux, Windows, Mac

Prerequisites: Fiji, Java 8 or newer

MPI-Ar-GE: SUMMARY

Labkit [1] is a Fiji [2] plugin for image segmentation. It provides an intuitive user interface for the interactive training and application of a random-forest-based pixel classification algorithm. We used Labkit to classify image pixels into foreground and background. The output of this is a sequence of binary images. A connected component analysis was subsequently applied in post-processing to obtain instance segmentations. Finally, we applied a dilation to grow the segments to the correct size.

MPI-Ar-GE: PREPROCESSING

In order to make use of Labkit's fast and memory-efficient data loading, we converted the images to the BigDataViewer XML+HDF5 file format [4] prior to processing. This was performed using the BigStitcher Fiji plugin [3].

MPI-Ar-GE: SEGMENTATION

The random forest based pixel classification algorithm is trained on a representative set of images. We used the frames 0, 1, 10, 20, 40, 50 and 59 from the first training sequence of **Fluo-N3DL-TRIF**. The Labkit user interface was used to manually annotate a few hundred pixels in those images as foreground or background. These manual annotations served as gold truth to train the pixel classifier. We intentionally only annotated the nuclei's central pixels as foreground. This causes the pixel classification algorithm to return segments that are smaller than the actual nuclei. Which makes it less likely for the segments of two nuclei to touch each other, minimizing the likelihood of segmentation errors due to under segmentation. The pixel classifier internally applies a list of image filters to extract pixel features. Those filters can be configured in a settings dialog. We used the following filters: *original image*, *gaussian blur*, *laplacian of gaussians*, and *hessian eigenvalues*, with the *sigma* values of 1, 2, 4, 8, and 16. Finally, we saved the trained pixel classifier to Labkit's classifier file format and used the Labkit command line tool [5] to run the classifier on both challenge sequences of **Fluo-N3DL-TRIF**.

MPI-Ar-GE: POST-PROCESSING

The output of the pixel classification is a binary mask. A connected component analysis was used to assign unique IDs to the individual segments. Due to the annotation procedure, the resulting foreground masks are smaller than the actual nuclei. Therefore, we dilated the segments to match the size of the nuclei. The dilation was done in three steps. The first two steps used a three-dimensional 6-neighborhood as dilation kernel. The third step used a 3×3×3 pixel cube as kernel. The combination of dilation kernels was chosen as it optimizes **SEG** on the training sequences of **Fluo-N3DL-TRIF**.

REFERENCES

1. <https://imagej.net/Labkit>
2. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: An open-source platform for biological-image analysis. *Nature Methods* **9**, 676-682 (2012).
3. Hörl D, Rusak FR, Preusser F, Tillberg P, Randel N, Chhetri RK, Cardona A, Keller PJ, Harz H, Leonhardt H, Treier M, Preibisch S. BigStitcher: Reconstructing high-resolution image datasets of cleared and expanded samples. *Nature Methods* **16**, 870-874 (2019).
4. Pietzsch T, Saalfeld S, Preibisch S, Tomancak P. BigDataViewer: Visualization and processing for large image data sets. *Nature Methods* **12**, 481-483 (2015).
5. <https://github.com/maarzt/labkit-command-line>