User guide: Platelet segmentation and classification

Jeremy Pike
j.a.pike@bham.ac.uk

Steve Thomas
s.thomas@bham.ac.uk

January 2020

This guide describes how to use the workflows presented by Pike *et al.* (2020) [1]. If you use or adapt these workflows you should cite this publication. The workflows make extensive use of the open-source software platforms KNIME[2] and ilastik[3, 4].

References

[1] Jeremy A Pike, Victoria A Simms, Christopher W Smith, Neil V Morgan, Abdullah O Khan, Natalie S Poulter, and Steven G Thomas. An adaptable analysis workflow for characterization of platelet spreading and morphology. *In press*, 2020.

[2] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

[3] Christoph Sommer, Christoph Straehle, Ullrich Köthe, and Fred A Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pages 230–233. IEEE, 2011.

[4] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, et al. ilastik: interactive machine learning for (bio) image analysis. *Nature Methods*, pages 1–7, 2019.

1 Installation

1. Download ilastik and install.
2. Download KNIME and install.
3. If you haven't already clone/download the workflows from our GitHub repository:

   https://github.com/JeremyPike/platelet-segmentation.

   This repository contains this user guide and the corresponding KNIME workflows.
4. Open KNIME and set the PlateletSegKNIME directory as the working directory. This directory holds all the KNIME workflows and should be selected whenever you want to use them.
5. Install and configure the KNIME Ilastik integration. Follow the linked instructions.

2 Generating training sets

To train the pixel classifier in ilastik, you will need to select some training images and convert them to a file format readable by ilastik. This is simple, but there are a few important considerations:

- Choose a subset of your data which encapsulates the full variability of the dataset. Images should come from across replicates and treatments and include the range of cell types you will experience in your samples.

- Make sure your training data is treated the same as your test samples (e.g. single slice, projection, etc.)

The "convertTrainingData" workflow generates and saves a series of cropped images as tif files which are compatible with ilastik.

Instructions

1. Create a directory to hold the training images (eg. trainingdata) and move your chosen training images to this folder.

2. Open KNIME and select the PlateletSegKNIME directory (not the individual workflows).

3. Once open, select the "convertTrainingData" workflow from the KNIME explorer.

4. Right click on the "set user defined parameters" node and select "Configure" (alternatively F6 or double click).

```
// system imports
// Your custom imports:

// system variables
// Your custom variables:

// expression start


// input data path
// eg. C:\\Users\\jeremy\\Documents\\data;"
out_inputPath =  "V:\\templateData\\trainingData";

// input file format
out_inputFormat = "czi";


// set number of slices above and below focus plane to project
// 0 chooses the single most in-focus slice, -1 will maximally project all slices
out_bandWidthProj = 2;
```

   There are three parameters to define:

   - out_inputPath should be set to the path of the directory containing your training files (sub-folders will be searched automatically). Note that you need to use \\ in the path name, enclose the path in double quotation marks and include a semicolon at the end of the line.
   - out_inputFormat specifies the format of your training files, for example "tif", "czi" or "nd2".
   - out_BandWidthProj specifies how z-stacks will be handled. If your images are 2D this will have no effect. Remember this needs to be the same for both the training data and the test data, so it might be worth confirming what works best for your images. Options include:
     -1 - will project all slices (standard maximal projection)
     0 - selects the single most "in-focus" slice
     1 – will create a projection of the most in-focus slice and the slice above and below this
     2 - will create a projection of the most in-focus slice and the 2 slices above and below this
     3 - will create a projection of the most in focus slice and the 3 slices above and below this

   Once complete, press OK to confirm the settings.

5. Image cropper node – to change options of how the images are cropped, right click on this node and click configure. On the options tab you can set the boundaries of the cropping regions. For training it is faster to use smaller cropped images. For example for 2048×2048 raw images use 769-1280 for both X and Y (this crops to a region in the middle of the image). If you want the whole FOV, select "all" for X and Y.

   For multi-channel images you can also choose which channels to use by configuring the Image reader node. In the "subset selection" tab select the channel(s).

6. Now click the "execute all executable nodes" button in the toolbar (Or press shift + F7). The workflow will run and will generate a series of tif files in the training data folder named "trainingImage_00", "trainingImage_01", etc. By chance the cropping may select a region of the image with little or no cells in it. If this happens regularly then choose different images for training or don't crop. To inspect the cropped images, right click on the image cropper node, select "image viewer" then double click on an individual image.

7. You can now close the workflow. If you save the workflow choose a different name to prevent overwriting the template. See Section 6 for further guidance on saving workflows.

3 Ilastik – building a pixel classifier

The pixel level classification is the critical step in being able to automatically segment the platelets in your data and time spent getting a good pixel classifier will be important in achieving good quality results.

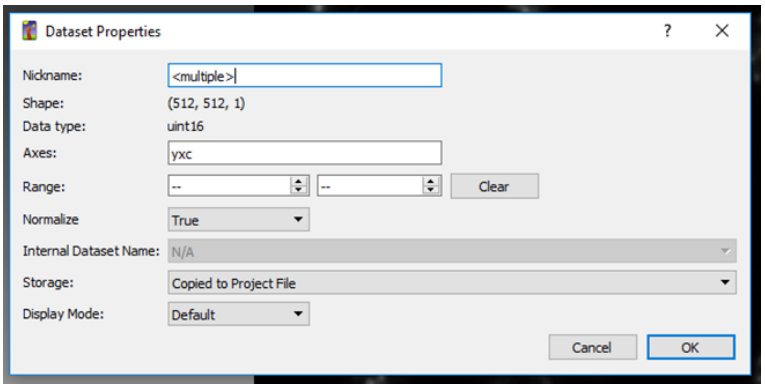
The classification works by the user telling the software what is cell and what is background. The algorithm then looks at many different features of the image and calculates how best to classify the categories you define. When the trained pixel classifier is then applied to new images, it uses this information to assign a cell or background class to each pixel. If needed further documentation on ilastik pixel classifiers can be found on the ilastik website.

Instructions

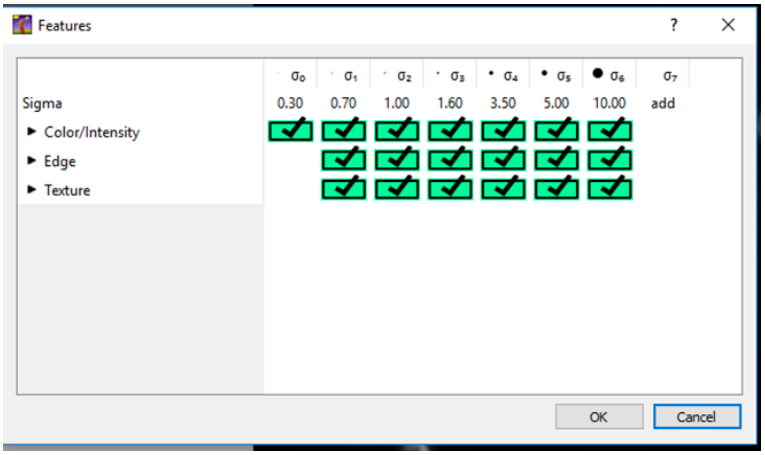
1. Launch ilastik and create a new project of type "Pixel classification", choose a name and location for the project file.

2. In the "raw data" tab, Click add data, then "add separate images". Browse to directory where you saved your training images generated using the "convertTrainingData" workflow and choose the images, click OK. Images are loaded.

3. Select all images, right click and select "edit shared properties". In the dialogue box that appears, under storage select "copied to project file". This will make your project file larger but mean that the training data is stored with the classifier.

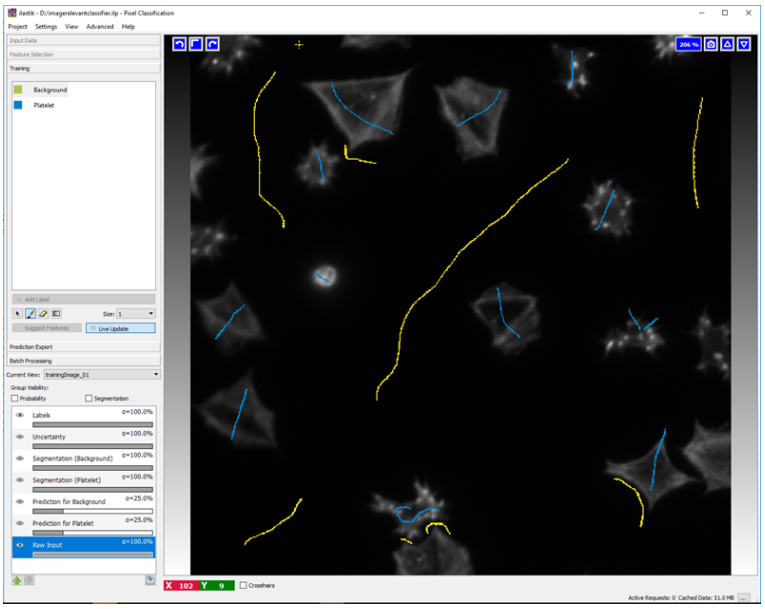


4. Click on "feature selection" tab and then click "select features" to bring up a feature selection dialogue. Selected all features as per picture. Click OK. Selecting all features is slower but will produce the best results.

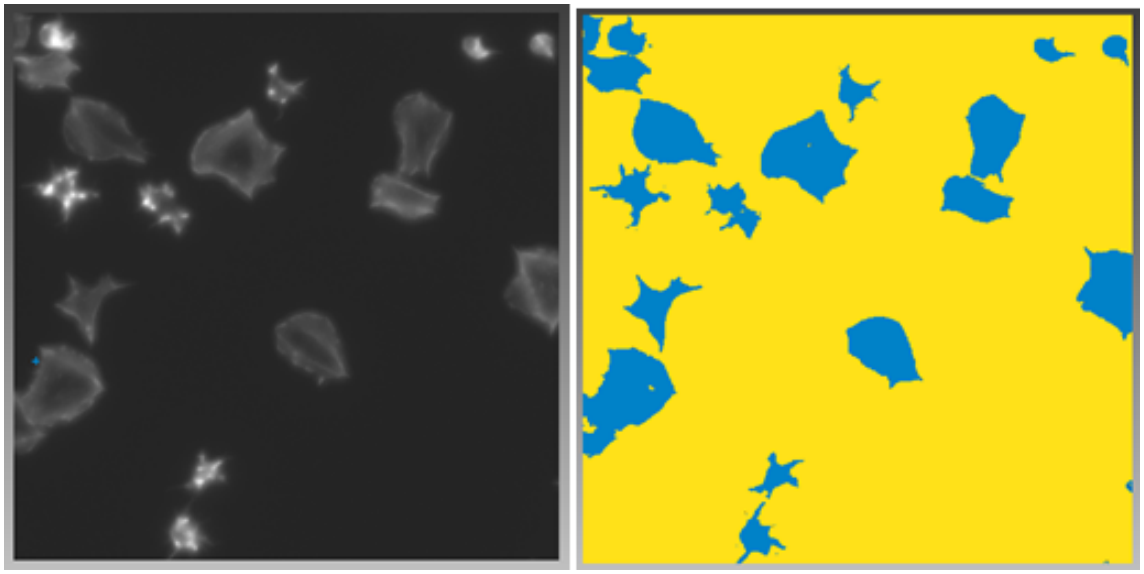


5. Select the "training" tab. Each label represents a classification. You can rename the labels so they make more sense to you. Label 1 should be defined as "platelets" and label 2 should be defined as "background" to ensure compatibility with later KNIME workflows.

6. Select a class (i.e. Background) and draw a line on the image that contains pixels of that type. The more accurately you do this the better the classification will be. To zoom in or out, hold the control button and use mouse wheel. You can also change the brush size by changing the number in the box labelled "size".



7. To see how well the classifier is working, click "live update". This will show you the classification applied to the current image. You can toggle between "probability" and "segmentation" to compare the image with the results. Continue to add "background" or "platelet" features to improve the segmentation. You can increase the intensity levels in the image by clicking the "levelling" button. Dragging the mouse left or right across the image will increase or decrease contrast. Right clicking in the image will reset the display settings. If you make a mistake with drawing a line, select the "eraser cursor" to rubout part, or all of a misplaced line.



8. You can move to the next image in the training series by using the "current view" box which will give a drop down menu of the available images. Continue defining "background" and "platelet" until the segmentation is working well across all the training images. Remember to go through all the images to ensure it works across all them all.

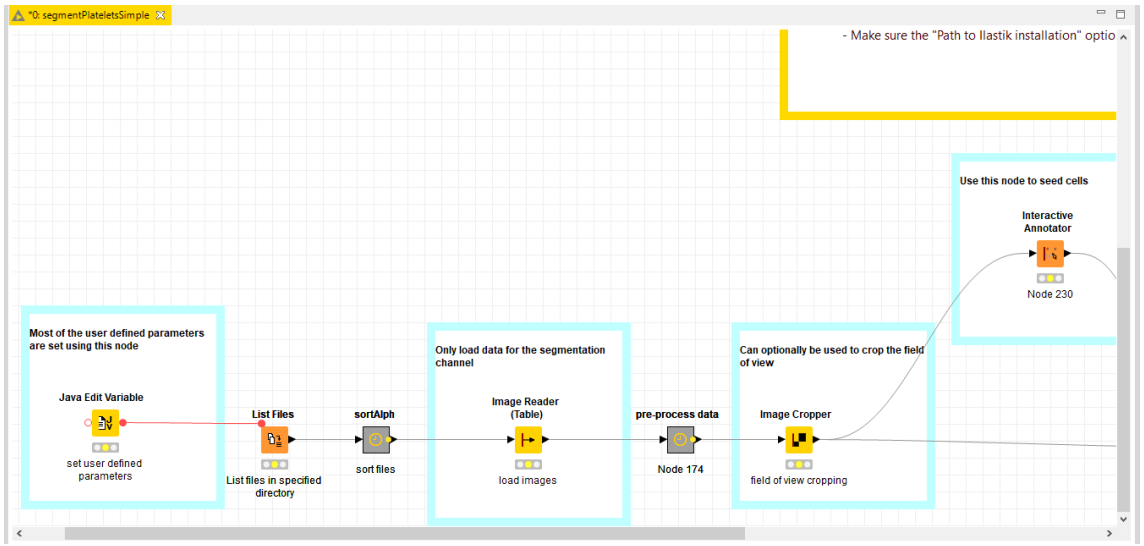
9. Once you are happy with the classifier, save the project and close ilastik.

4 Running the workflow on a full data-set

There are two main workflows you can run. "segmentPlateletSimple" will produce per platelet measurements including spread area, whereas the "segmentPlateletsObjectClassification" workflow can additionally classify platelets into groups based on morphology. In this section we we cover the "segmentPlateletSimple" workflow.

Instructions

1. Open the "segmentPlateletsSimple" workflow. To retain the default workflow intact, make a new copy of the workflow for this dataset using "File – Save As". Scroll down the main window so you can see the workflow nodes.



2. Right click on the "set user defined parameters" node and select configure. This opens a dialogue box with a number of parameters to configure:

```
// system imports
// Your custom imports:

// system variables
// Your custom variables:

// expression start

// path to ilastik project file
//e.g C:\\Users\\jeremy\\Documents\\pixelClassifier.ilp
out_ilastikProjectPath = "V:\\templateData\\pixelClassifier.ilp";

// input data path
// eg. C:\\Users\\jeremy\\Documents\\data"
out_inputPath =  "V:\\templateData\\testImages";

// input file format
out_inputFormat = "czi";

// output excel file name and location
// eg. C:\\Users\\jeremy\\Documents\\output.xlsx"
out_outputFile =  "V:\\templateData\\output.xlsx";

// click all cells (1) or only touching (0)
out_allCells = 1;

// set number of slices above and below focus plane to project
// 0 chooses the single most in-focus slice, -1 will maximally project all slices
out_bandWidthProj = 2;

// set threshold for probability map (normally 0.5)
out_probabilityMapThreshold = 0.5;

// set minimum platelet area and maximum hole size in microns
out_minPlateletAreaMicrons = 4.;
out_maxHoleSizeMicrons = 10.;
```

- out_ilastikProjectPath sets the path to the ilastik classifier you have trained for these images. Remember to enclose the path in " ", use \\ in the pathname and put a semicolon at the end of the line. Include the .ilp extension at the end of the classifier filename.
- out_inputPath sets the path to the directory containing the input data (not the training data but the files you want to analyse).
- out_inputFormat specifies the format of your data files, for example "tif", "czi" or "nd2".
- out_outputFile sets the output file path and name (**this should have a .xlsx extension**)
- out_allCells specifies which seeding strategy you are going to use:

  1 - seed all cells

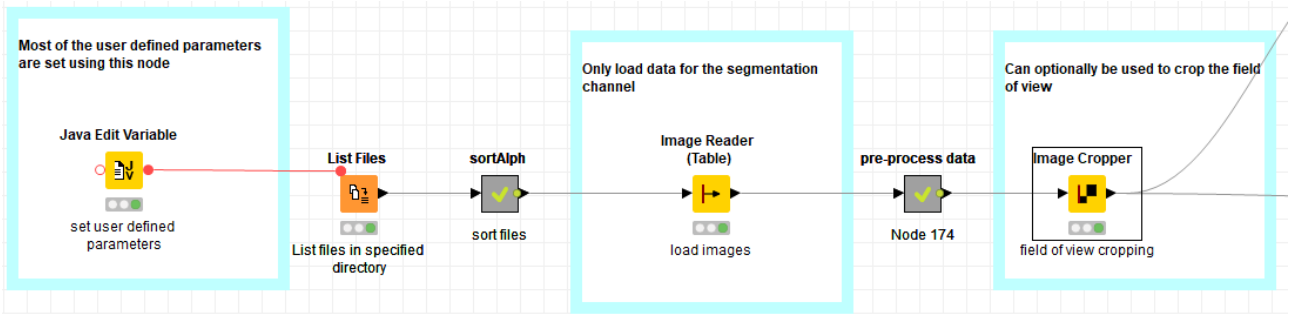
  0 - seed only touching cells
- out_BandWidthProj specifies how z-stacks will be handled. This should be the same value as used to prepare the training data.
- out_probabilityMapThreshold Specifies the threshold for the probability map produced by ilastik. Default value of 0.5 is okay for most applications.
- out_minPlateletAreaMicrons Objects smaller than this minimum area are discarded. Calibrated to the same units as the input data.
- out_maxHoleSizeMicrons Holes in the segmentation smaller than this value are filled in. Calibrated to the same units as the input data.

Click OK to confirm the settings. The dialogue box will close

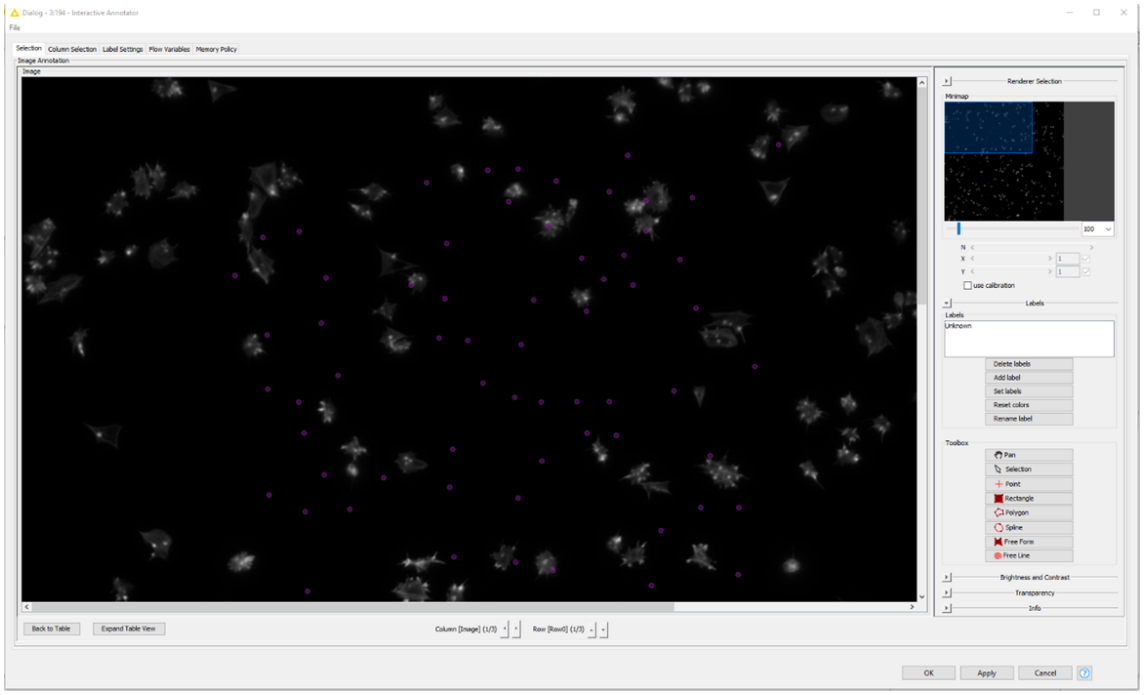
3. Configure the Image cropper node to choose how images are cropped (right click on this node and click configure). You don't need to crop to the same size as the training data. You will only need to crop images at this stage if you have too much data!

   For multi-channel images you can also choose which channels to use by configuring the Image reader node. In the "subset selection" tab select the channel(s).

4. Execute up to the image cropper node (right click on the "image cropper node" and select execute). The first 5 nodes will run and return green lights if successful:



5. Now configure the interactive annotator node which will open a window as shown below:



This shows the 1st image of your dataset in the main window and some simple tools down the right hand side. If there are any seeds left from a previous analysis or the template data clear them by selecting the "unknown" class in the labels box and clicking "delete labels".

6. Now you need to manually place seeds on all the images in your dataset. Note that seeding is the slowest part of the process. However seeds can be saved and modified as required, so once the process is complete it does not need to be repeated, even if you change the classifier. You can also use the same seeds within the object classification workflow if required. This can be done by copying the annotator from this workflow to the other. There are two ways to place seeds as specified with the "out_allCells" variable:

Option 1: Click on all cells (out_allCells = 1)

- Use the point tool to place a seed at the centre of each cell.
- If you don't want to include an object simply don't click on it. In a clump, seed all cells, or none at all. Don't only seed some.

Option 2: Click only on touching cells (out_allCells = 0)

- Use the point tool to place a seed at the centre of cells which are touching other cells.
- If two platelets are very close together but not touching click on them anyway as the ilastik classifier may not be able to separate them.
- To exclude objects create a label called "discard" (use the "Add label" button) and use this to select objects which you don't want to include. Remember to return to the default "Unknown" class after.

Things to note for both options:

- Try to get seeds close to the centre of each cell.
- Cells that touch the edge of the image will be ignored during the post-processing and can be optionally seeded. However, if a cell is touching a cell which is touching the edge, BOTH cells need to be seeded for the process to work correctly.
- If you make a mistake with a seed, right click on the seed to remove it.
- You can alter the brightness/contrast of the image to make it easier to see faint cells.
- Point colours can be changed for better contrast ("Reset colors").

7. After each image is complete move to the next image by clicking the down arrow labelled "Row" below the main image window. Work through all of your images. It is worth saving regularly to avoid losing seeds in event of a crash. To do this close the Interactive Annotator by pressing "ok" then save the workflow.

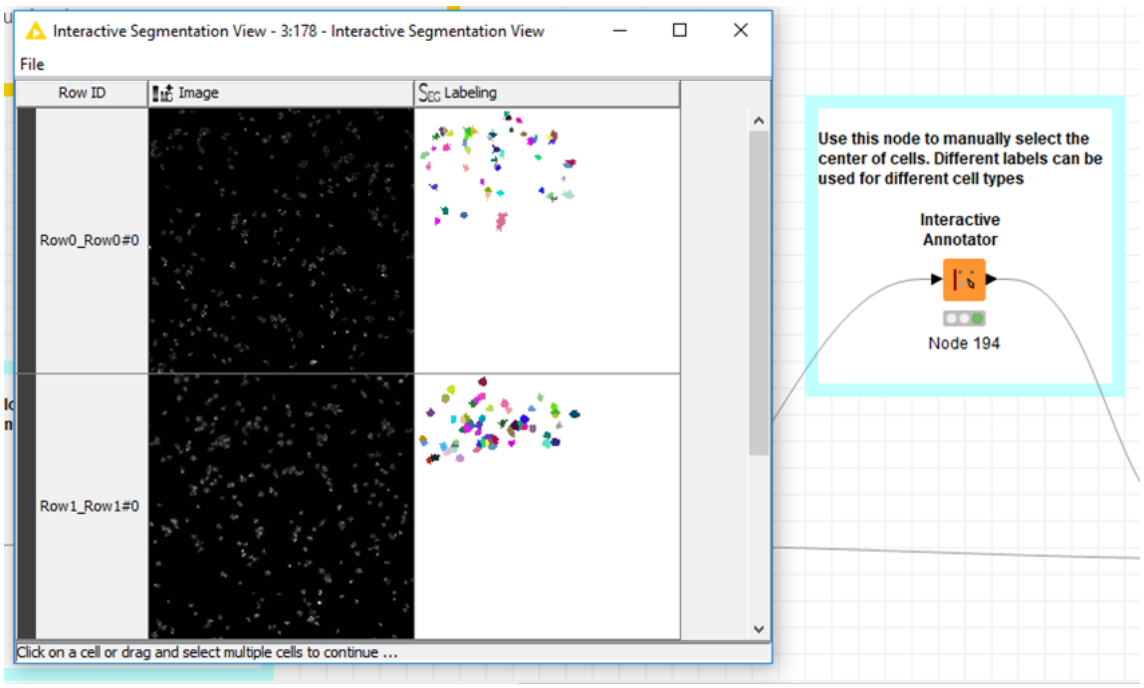
   Note – there is currently a bug in KNIME which means that sometimes the image annotator node doesn't update when you have cropped images differently. If this error occurs then you need to delete the image annotator node and insert a new one. Instructions on how to do this are in the right hand yellow box above the workflow. If you have changed any crop settings in the workflow it is worth adding a new annotator BEFORE doing the seeding as if you have to do it afterwards you will lose your seeds!

8. The remainder of the workflow is ready to run. To do this click "run all" (or shift + F7) on the top menu bar. The workflow will now run – the time for this will vary depending on the number of images and seeds but it can take some time to run the ilastik pixel classifier. As each node completes the light will turn green. Errors will be indicated in red.

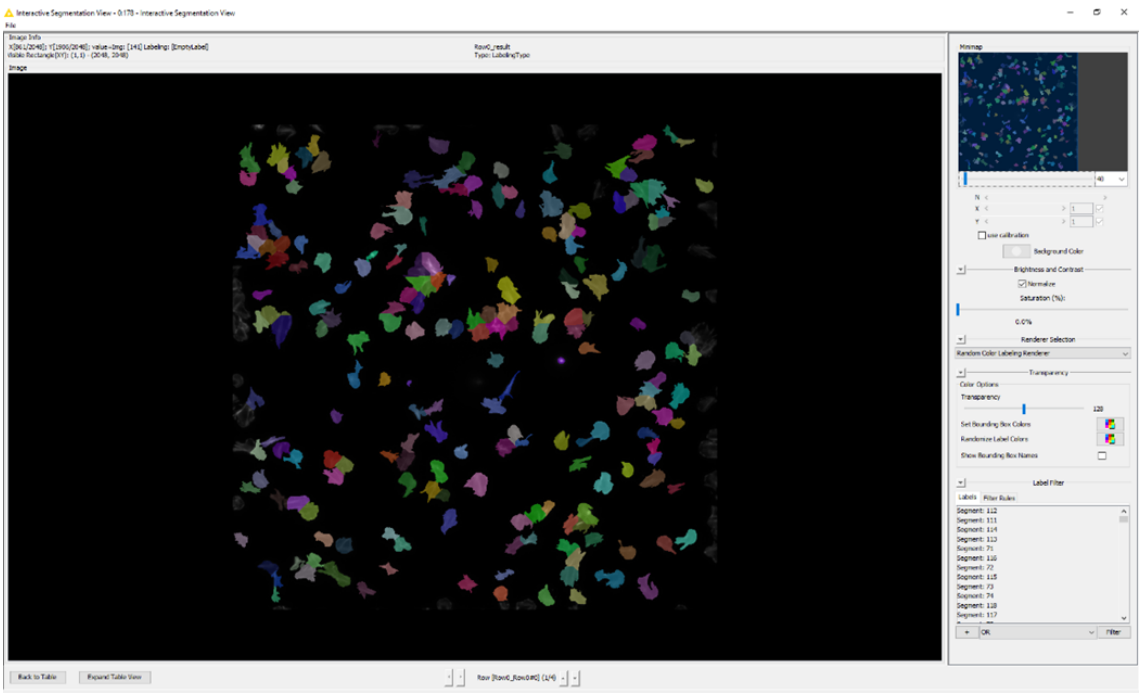
   Note – KNIME needs to know where your ilastik installation is located. If you get an error at this point check this is configured properly. Instructions on how to do this are in the right hand yellow box above the workflow.

9. Once the workflow has finished an excel file containing per platelet statistics will have be saved.

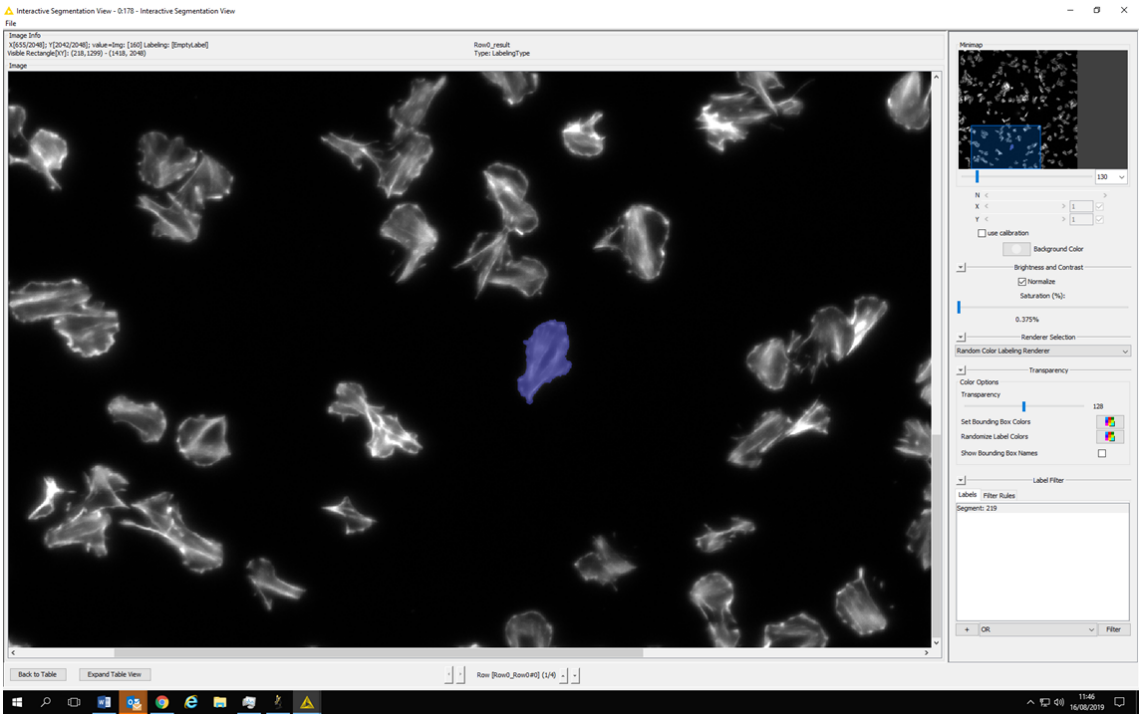
10. You can inspect the segmentation results using the "Interactive segmentation view" node. After this node is executed right click and select "View:Interactive Segmentation View". This will open a new window which contains a table containing the original images and the output from the segmentation workflow.



Double clicking on a row will open it in the viewer and you can investigate the success (or failure!) of the segmentation process.



Individual cell segmentations can be viewed by selecting a segment (or a number of segments) in the Label Filter box (bottom right), right clicking and selecting "filter selected". This will show only the selected segment. Changing the brightness and contrast saturation and transparency slides will allow you to establish the success of the segmentation. Filters can be cleared by right clicking and selecting clear filters.



There are a couple of common reasons why the segmentation performance might be poor:

(a) The pixel classifier is not working reliably for a subset of images. This well be evident as the cells will not be well separated from the background. To correct this is often helps to add additional training labels or perhaps additional training images (done within ilastik). Once the ilastik pixel classifier has been updated and saved, you can re-run the workflow with the new classifier (everything else should be kept the same, no need to re-seed!) to test its success.

(b) Multiple cells are segmented as one cell. This is simply because seeds have been missed by the user. To correct go back to the interactive annotator and add the necessary seeds. Once seeding is updated the latter stages of the workflow will need to be re-run.

5 Object classifcation

This section covers classification of platelets into predefined classes. To do this we must train an object classifier and then run this on the full dataset.

Instructions

1. Choose a subset of images to train the object classifier on. These should contain examples of all platelet types and be chosen across replicates and conditions. In most cases the same images which were used to train the pixel classifier are sufficient.

2. Open the "trainObjectClassifier" workflow. As before, save to a new name to preserve the default workflow.

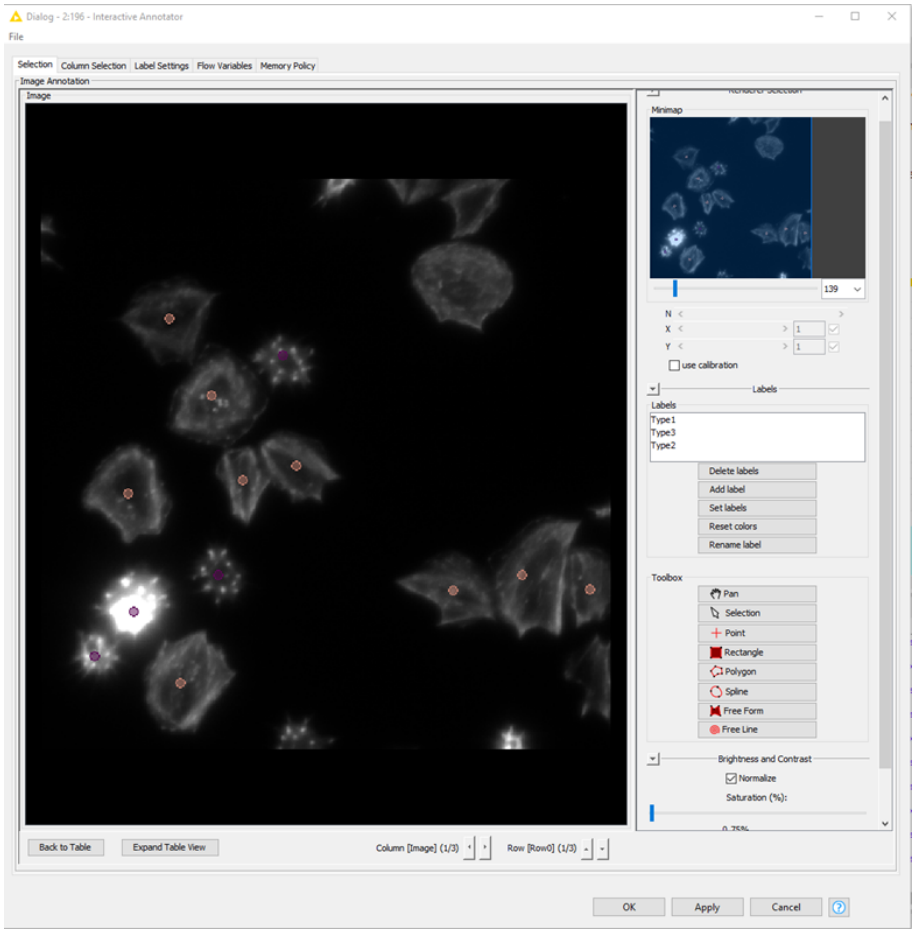
3. Configure the "set user defined parameters" node:

   - out_ilastikProjectPath sets the path to the ilastik classifier you have generated for these images. Remember to enclose the path in " ", use \\ in the pathname and put a semicolon at the end of the line. Include .ilp extension at the end of the classifier filename.
   - out_inputPath sets the path to the directory containing the training data (not the full dataset).
   - out_inputFormat specifies the format of your data files, for example "tif", "czi" or "nd2".
   - out_outputFile sets the output file path and name (**this should have a .zip extension**).
   - out_BandWidthProj Defines our z-stacks are processed. This should be the same value as used to prepare the training data.
   - out_probabilityMapThreshold Specifies the threshold for the probability map produced by ilastik. Default value of 0.5 is okay for most applications.
   - out_minPlateletAreaMicrons Objects smaller than this minimum area are discarded. Calibrated to the same units as the input data.
   - out_maxHoleSizeMicrons Holes in the segmentation smaller than this value are filled in. Calibrated to the same units as the input data.

4. Configure and execute the Image cropper node. For multi-channel images you can also choose which channels to use by configuring the Image Reader node.

5. Open the Image Annotator node. First create a label for each class you want to classify using the "Add label" button, give each label a meaningful name ("Rename label"). The colour for each label can be reset using the "Reset colors button".

6. Proceed to seed cells using the label that matches the cell type and the point tool. For this workflow seed all cells (not only those that are touching). Work through all the training images, saving regularly. It is important to be realistic about the classes you choose. Classes which are only subtly different from each other may not be robustly distinguishable.



7. Execute all remaining nodes (shift + F7). This will segment the cells, train the classifier and save it as a zip file in the specified location.

8. Twenty percent of the training labels are reserved for performance evaluation of the classifier. To visually inspect the classification results on the evaluation set use the Image Viewer node after the "createClassLabellings" Metanode.

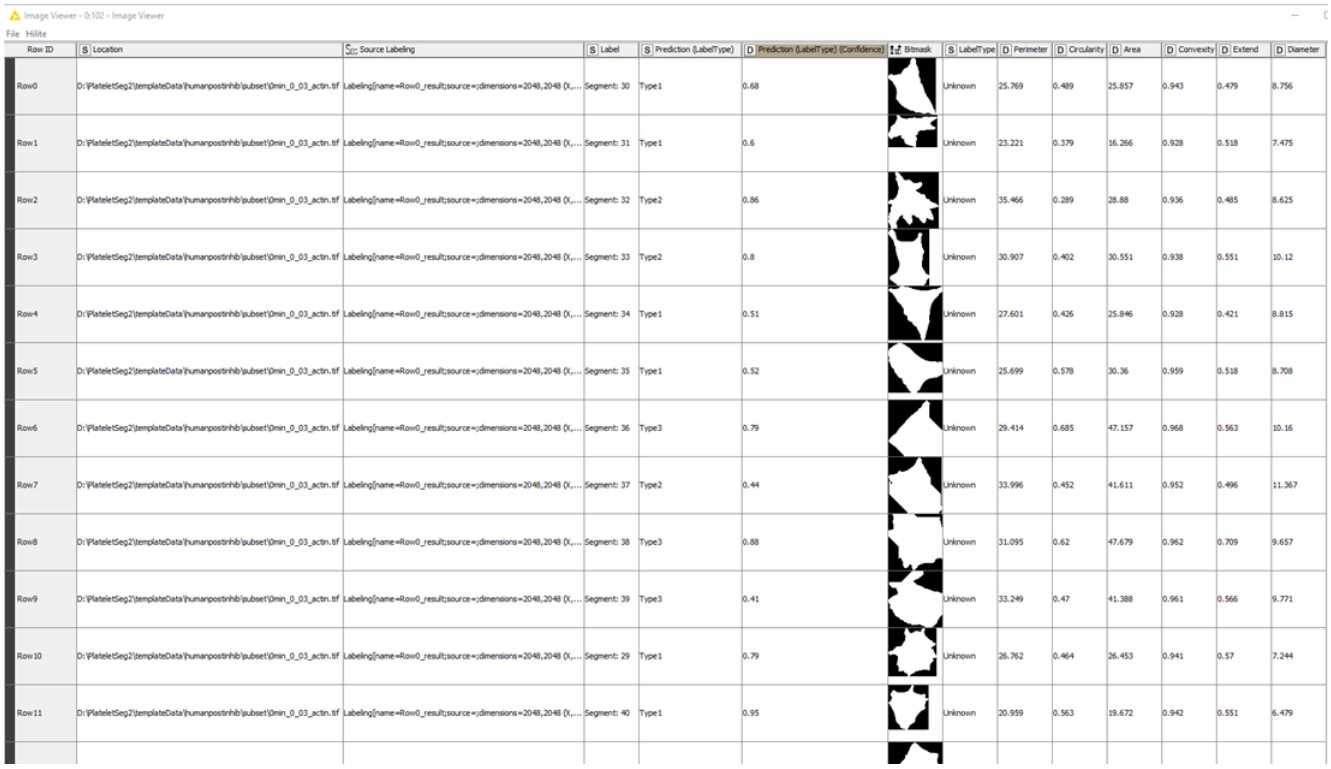
9. The performance of the classifier is reported in through the "Scorer" node (right click and select "Accuracy Statistics") and visualised though a confusion matrix using the "HeatMap" node (right click and select "View: Heat Map"). If performance is poor consider either adding additional training data or using less classes.

10. Save and close the training working and open the "segmentPlateletsObjectClassification" workflow. As before, save to a new name to preserve the default workflow.

11. Configure the set user defined parameters node. This is the same as the "segmentPlateletsSimple" workflow (Section 4), with the addition of the path for the object classifier you have generated.

12. Seed cells as per the "segmentPlateletsSimple" workflow and then run all remaining nodes (shift + F7). Note the seed do not require a class label as this will be assigned by you object classifier. If you have previously seeded the cells using the "segmentPlateletsSimple" workflow then copy the "Interactive annotator" node from the old workflow and replace the corresponding node in currently workflow.

13. An excel file containing per platelet statistics will have been exported to the specified path. To visualise this table within KNIME right click on the "Random Forest Predictor" node and select "Prediction output". This will open a table containing each of the segmented cells. In addition to the measurement parameters derived from the segmentation, the table will also include a prediction of the type of cell and confidence in that prediction.



Row ID	Location	Source Labeling	Label	Prediction (LabelType)	Prediction (LabelType) (Confidence)	Bitmask	LabelType	Perimeter	Circularity	Area	Convexity	Extend	Diameter
Row0	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 30	Type1	0.68		Unknown	25.769	0.489	25.857	0.943	0.479	8.756
Row1	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 31	Type1	0.6		Unknown	23.221	0.379	16.266	0.928	0.518	7.475
Row2	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 32	Type2	0.86		Unknown	35.466	0.289	28.88	0.936	0.485	8.625
Row3	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 33	Type2	0.8		Unknown	30.907	0.402	30.551	0.938	0.551	10.12
Row4	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 34	Type1	0.51		Unknown	27.601	0.426	25.846	0.928	0.421	8.815
Row5	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 35	Type1	0.52		Unknown	25.699	0.578	30.36	0.959	0.518	8.708
Row6	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 36	Type3	0.79		Unknown	29.414	0.685	47.157	0.968	0.563	10.16
Row7	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 37	Type2	0.44		Unknown	33.996	0.452	41.611	0.952	0.496	11.367
Row8	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 38	Type3	0.88		Unknown	31.095	0.62	47.679	0.962	0.709	9.657
Row9	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 39	Type3	0.41		Unknown	33.249	0.47	41.388	0.961	0.566	9.771
Row10	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 29	Type1	0.79		Unknown	26.762	0.464	26.453	0.941	0.57	7.244
Row11	D:\PlateletSeg2\templateData\humanpostinhib\pubset\0min_0_03_actin.tif	Labeling[name=Row0_result;source=;dimensions=2048,2048 (X,...	Segment: 40	Type1	0.95		Unknown	20.959	0.563	19.672	0.942	0.551	6.479

To visualise the classification results you can also using the Image viewer node (right click and and select “View: Image Viewer”).

6 Workflow saving and archiving

There are two options:

1. The workflow can be saved with the nodes executed (in green). This is not recommended as it can result in very large files as all the images, segmentations and intermediate steps are saved with the workflow. However it is useful if you need to quickly go back and check the results.

2. For large projects and for longer term archiving, it is best to reset all the nodes (right click on first node and select reset). This will retain the seeds, but not keep copies of the images, outputs etc and will result in a small file size. If needed the results can be re-generated by re-running the workflow. The caveat to this is that the directory structure and filenames for the raw data should not be changed as they will need to be reloaded in the same order to match the seeds. For archiving it is good practice to save a copy of the workflow, the pixel classifier, the object classifier (if using one) and the input data together.