PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

Machine Learning for Pixel and Object Classification

Robert Haase

With Material from

Deborah Schmidt, Jug Lab, MPI CBG

Uwe Schmidt, Myers Lab, MPI CBG

Martin Weigert, EPFL

Ignacio Arganda-Carreras, Universidad del Pais Vasco

@haesleinhuepf

July 2021

Lecture overview

Today

- Machine learning for Pixel and Object Classification
  - Random Forest Classifiers
  - Algorithm evaluation
  - Outlook: Deep learning
- Fiji
  - Trainable Weka Segmentation
  - Labkit
  - CLIJ object classifier
- Python
  - scikit-learn / napari

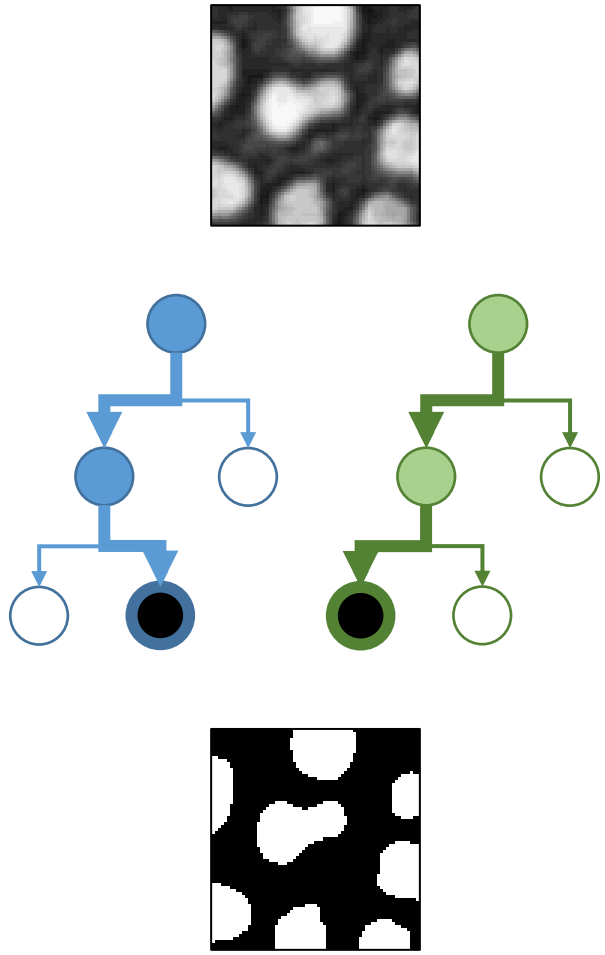


Image segmentation using thresholding

- Recap: Finding the right workflow towards a good segmentation takes time

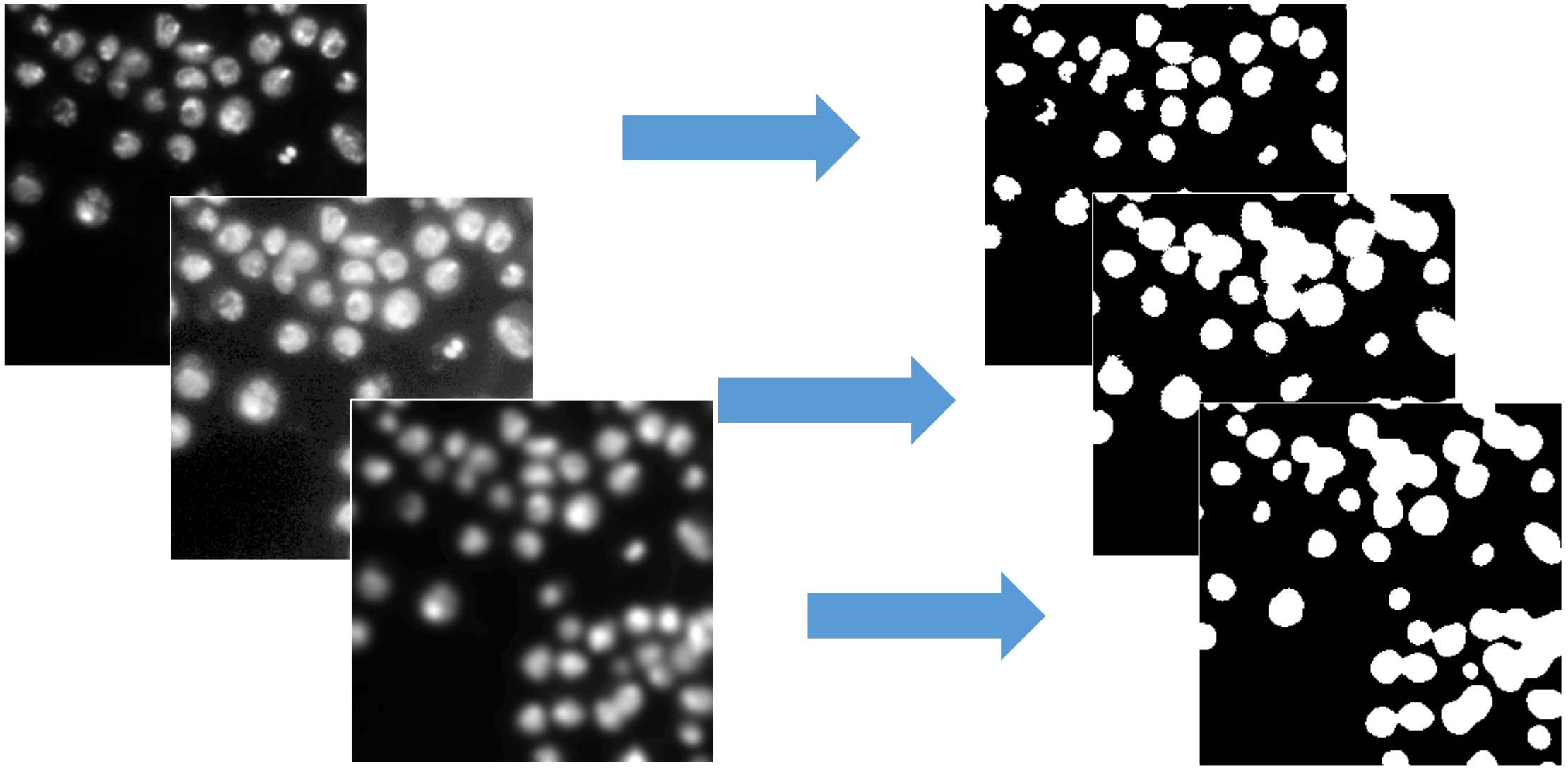


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Image segmentation using thresholding

- Recap: Combining images, e.g. using Difference of Gaussian (DoG)

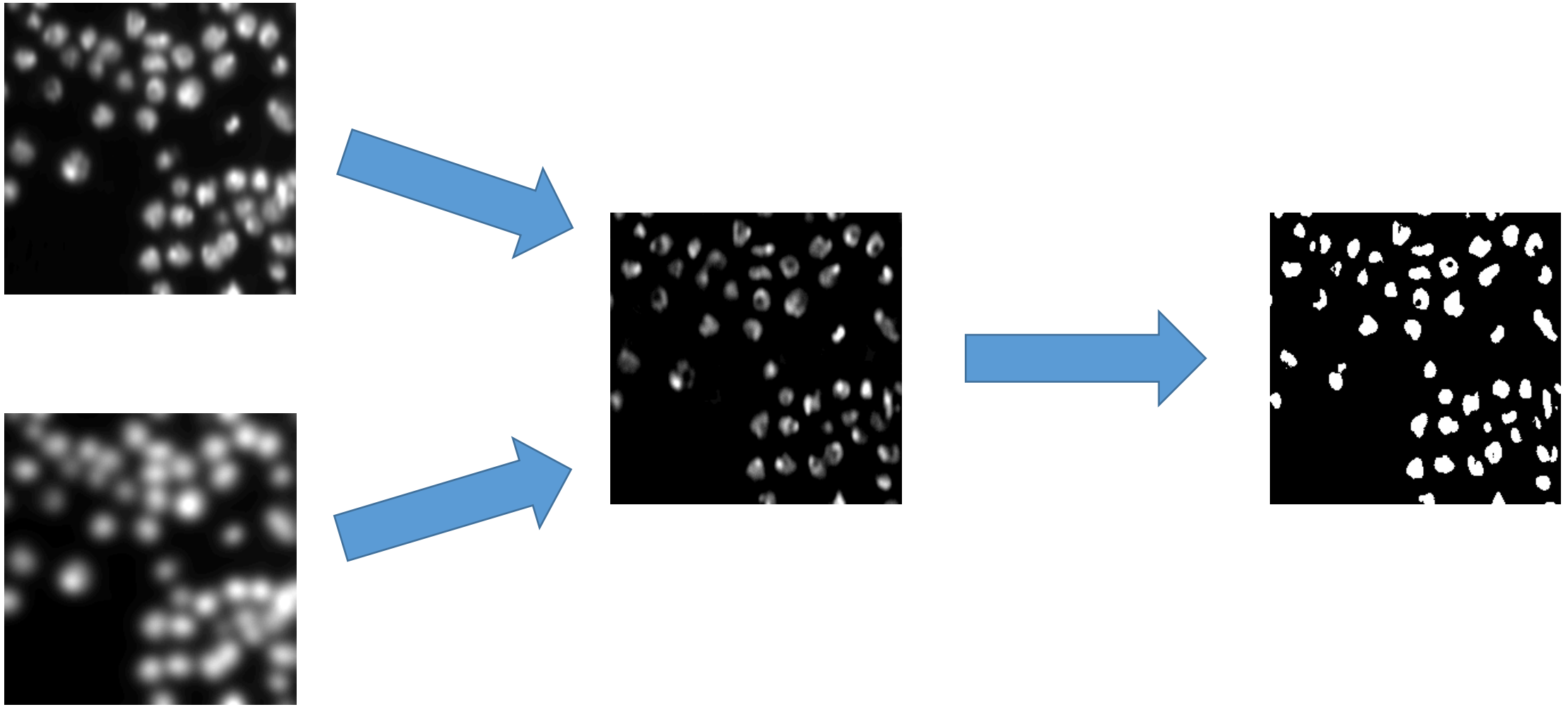


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Image segmentation using thresholding

- Might there be a technology for optimization which combination of images can be used to get the best segmentation result?

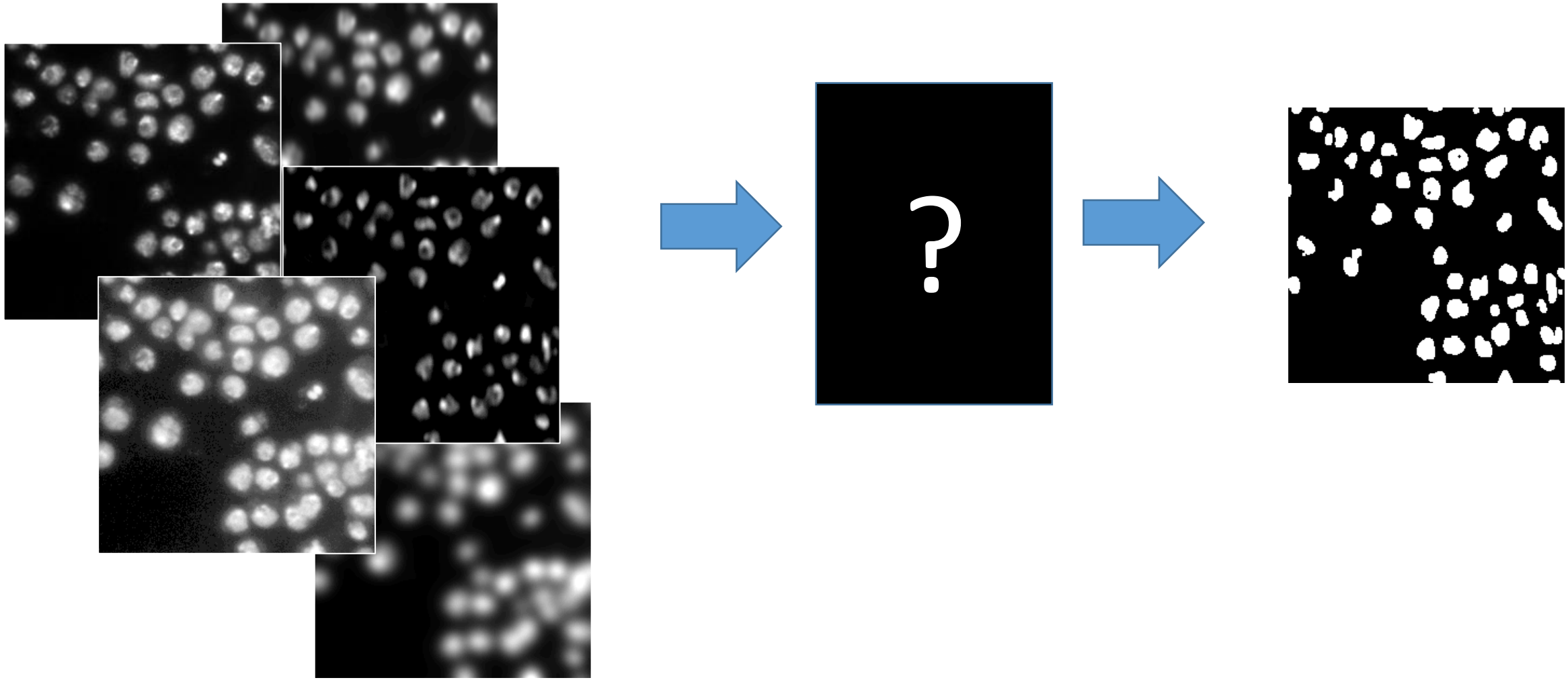
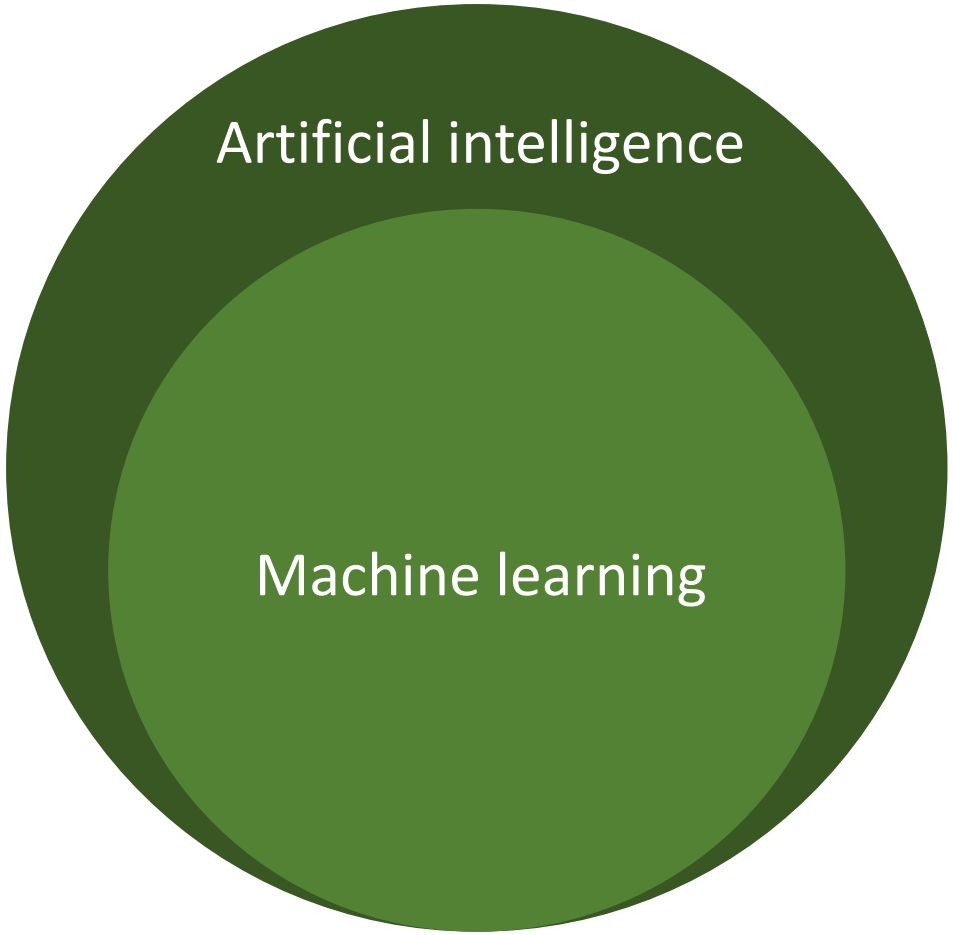
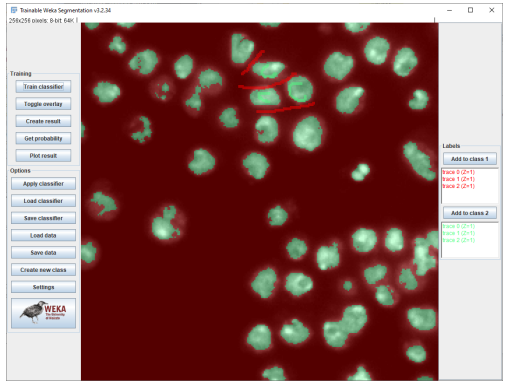


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

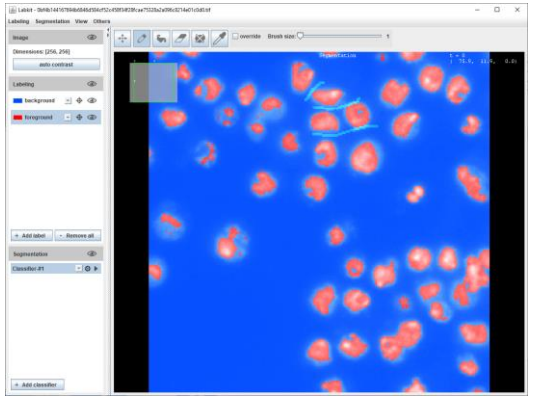
Machine learning

- A research field in computer science
- Finds more and more applications, also in life sciences.





Trainable Weka Segmentation



LabKit

Python /
scikit-learn /
napari

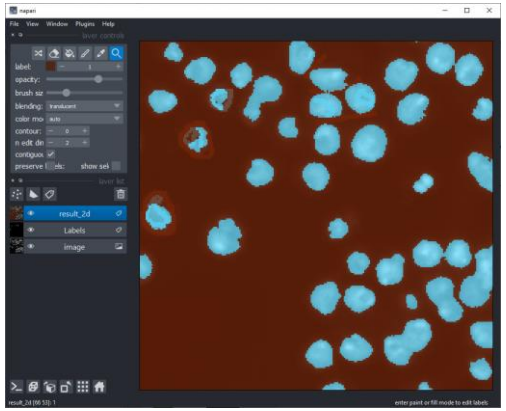
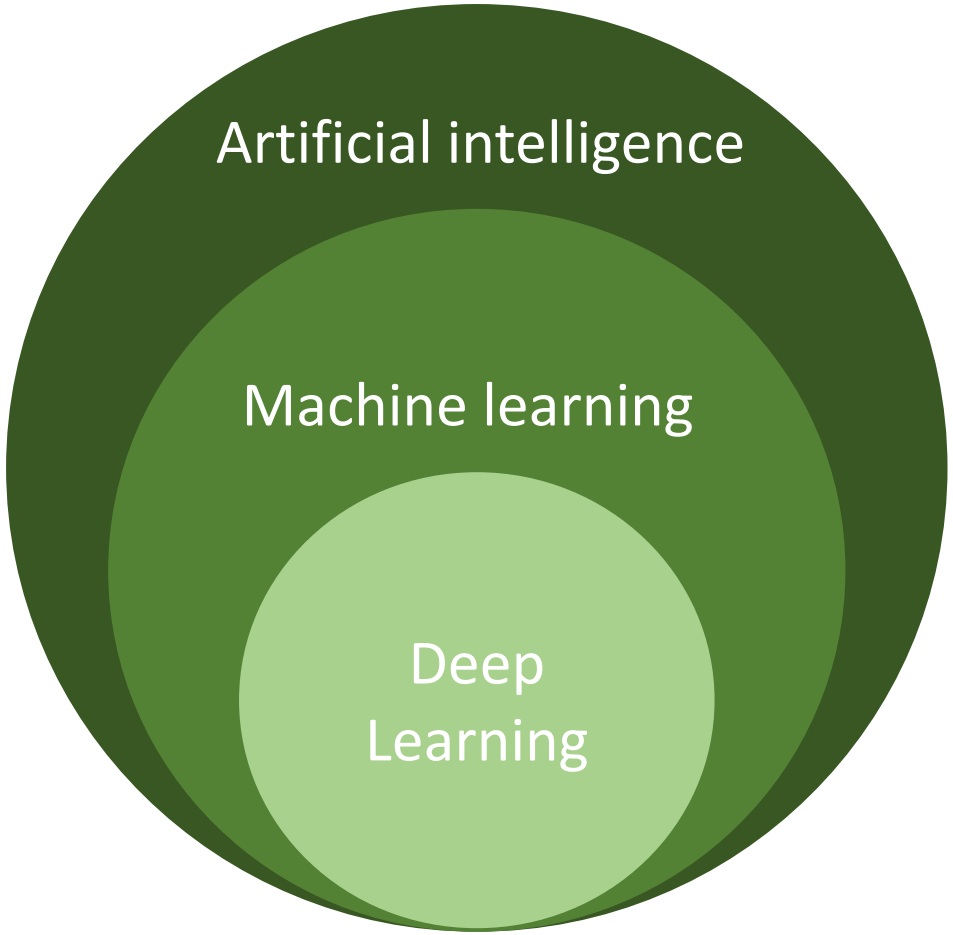


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Machine learning

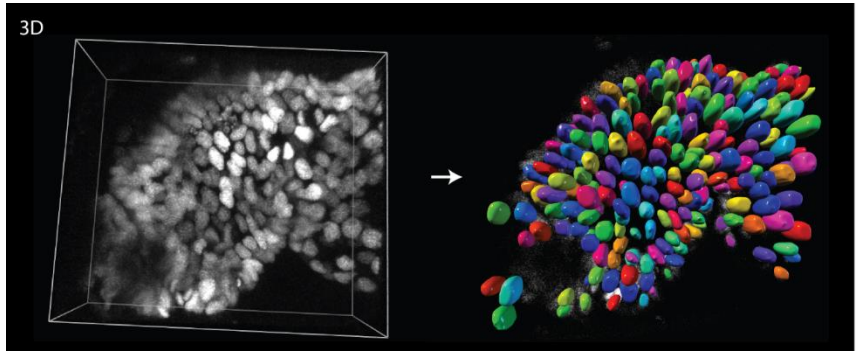
- A research field in computer science
- Finds more and more applications, also in life sciences.



deepImageJ

https://deepimagej.github.io/deepimagej/

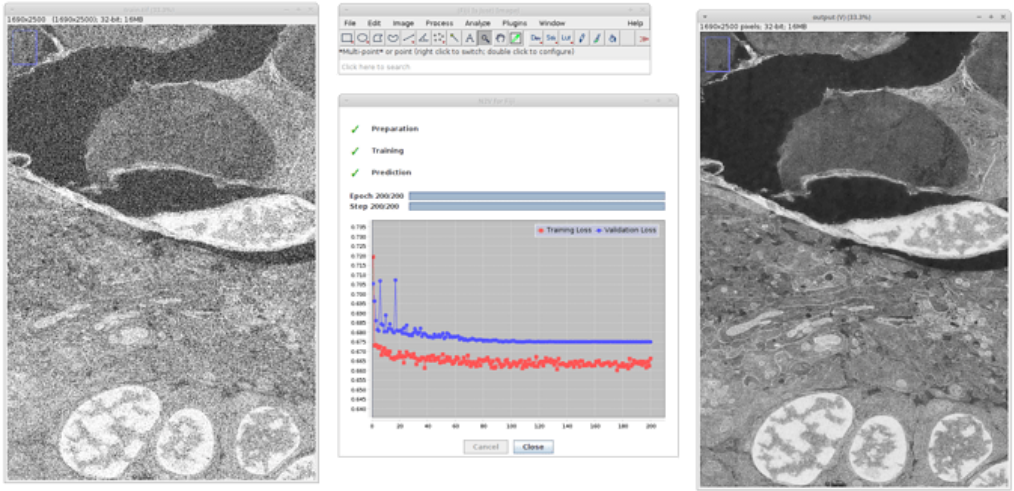
StarDist



https://imagej.net/StarDist

CSBDeep

Noise2Void

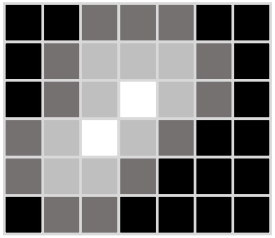


https://forum.image.sc/t/noise2void-for-fiji/34552

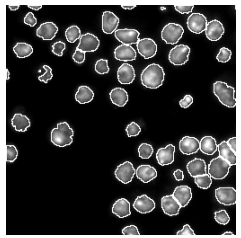
Machine learning

- Automatic construction of predictive models from given data

Pixels,

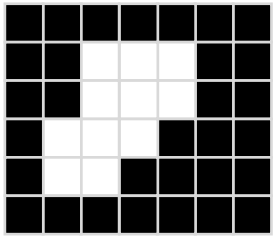


Objects,



Images, Audio, Text, Measurements, …

Dense Segmentation / Binarization



Object classification

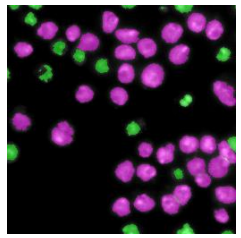
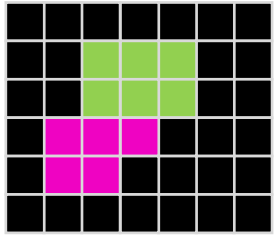


Image classification

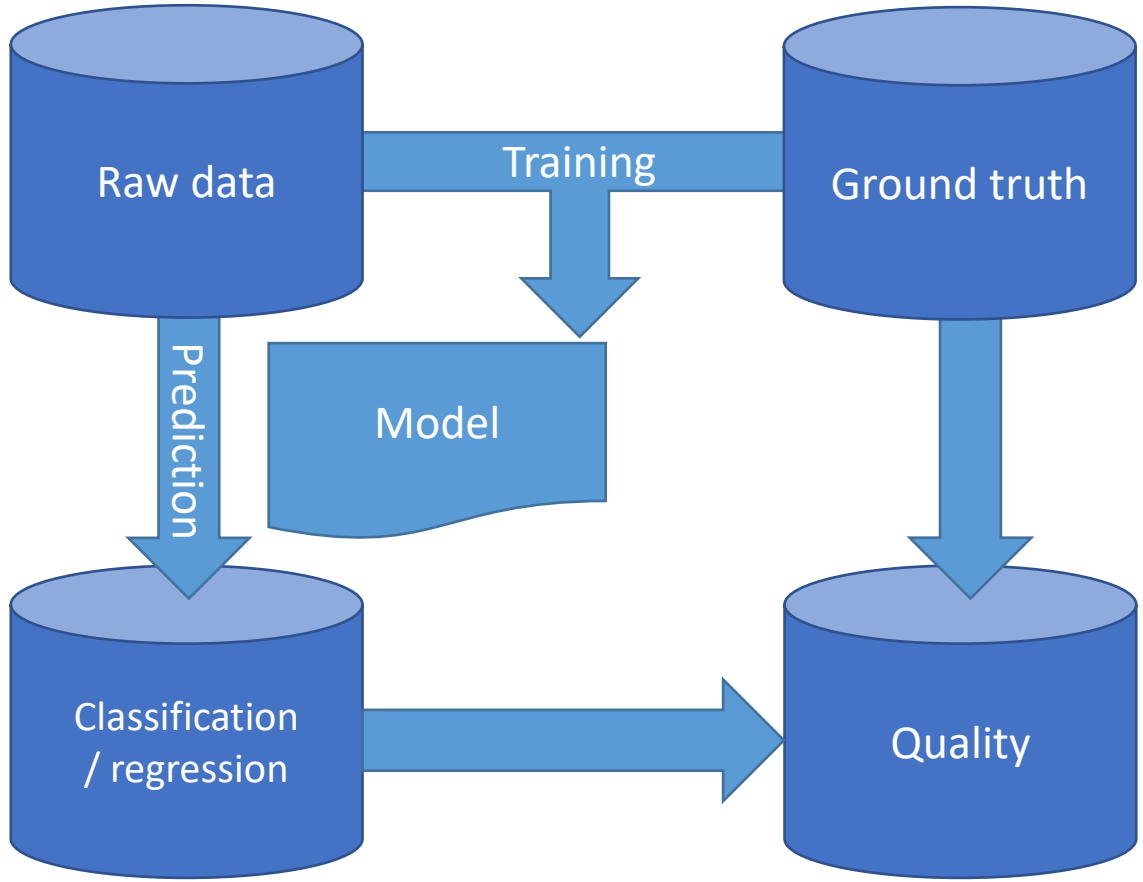
"Cat"

Instance segmentation



Cont. quantity

P_{Cat}= 0.5
$P_{Microscope}$= 0.4
Height = 80 cm



Annotated raw data, usually generated by humans

Precision, Recall

Overfitting / abstraction – selection of trees for the forest

- A good classifier is trained on a hand full of datasets and works on thousands similarly well.
- In order to assess that, we split the ground truth into two set
  - Training set (50%-90% of the available data)
  - Test set (10%-50% of the available data)

Typically done with hundreds or thousands of cells / images / objects / whatever.

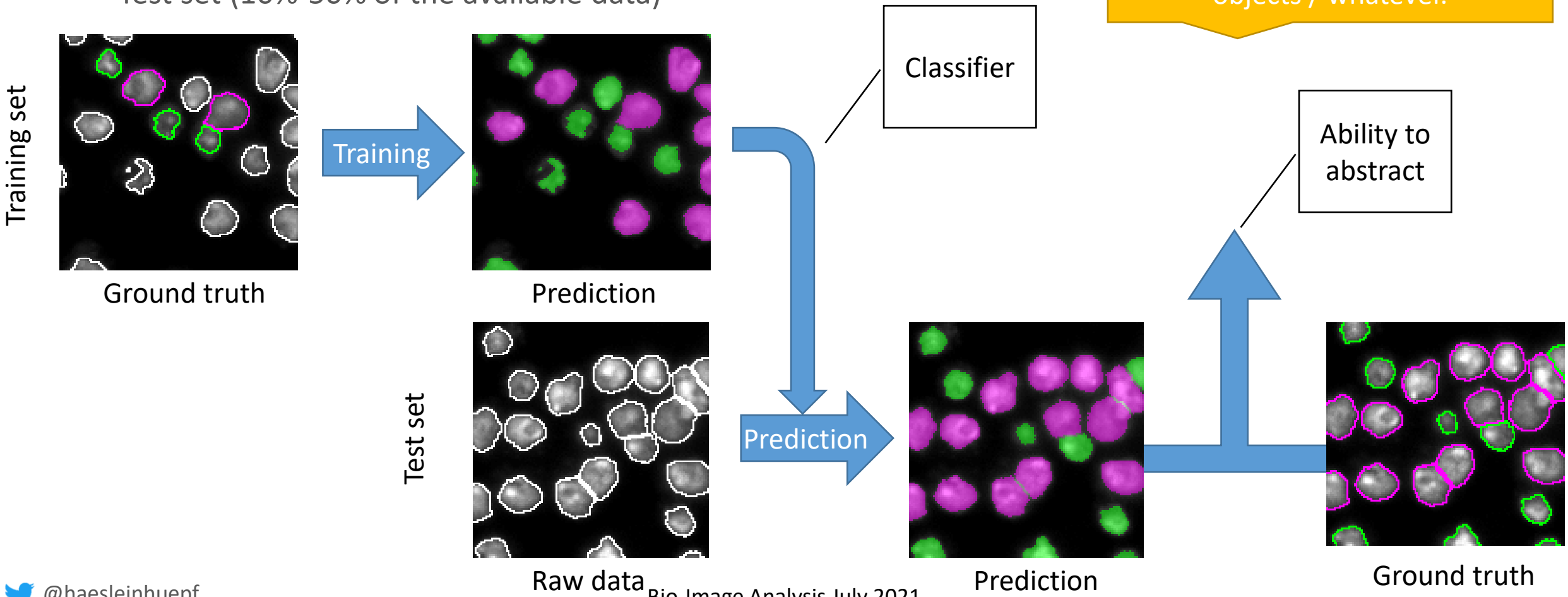
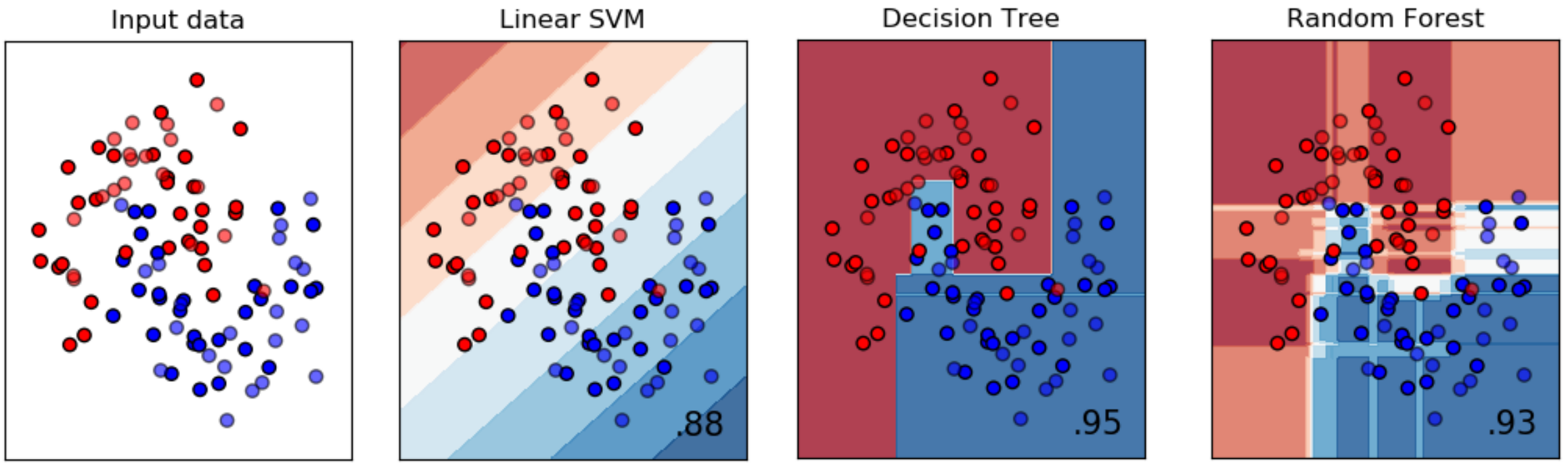


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Goal

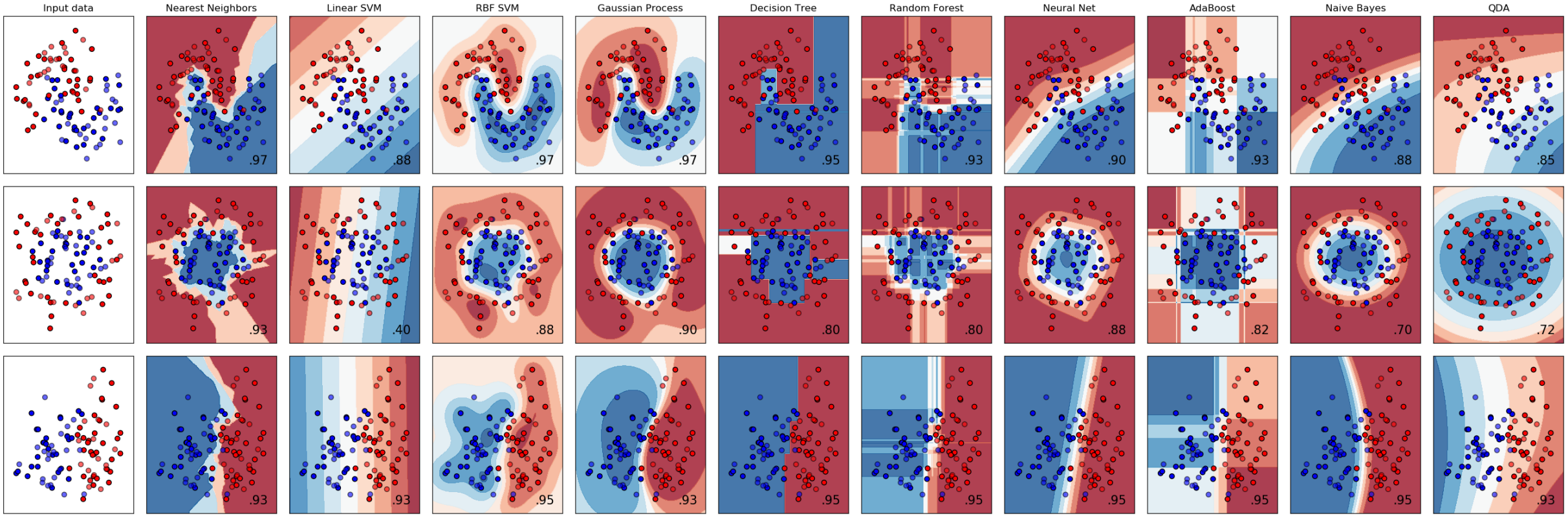
- Guess classification (color) from position of a sample in parameter space.



Adapted from https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Approaches

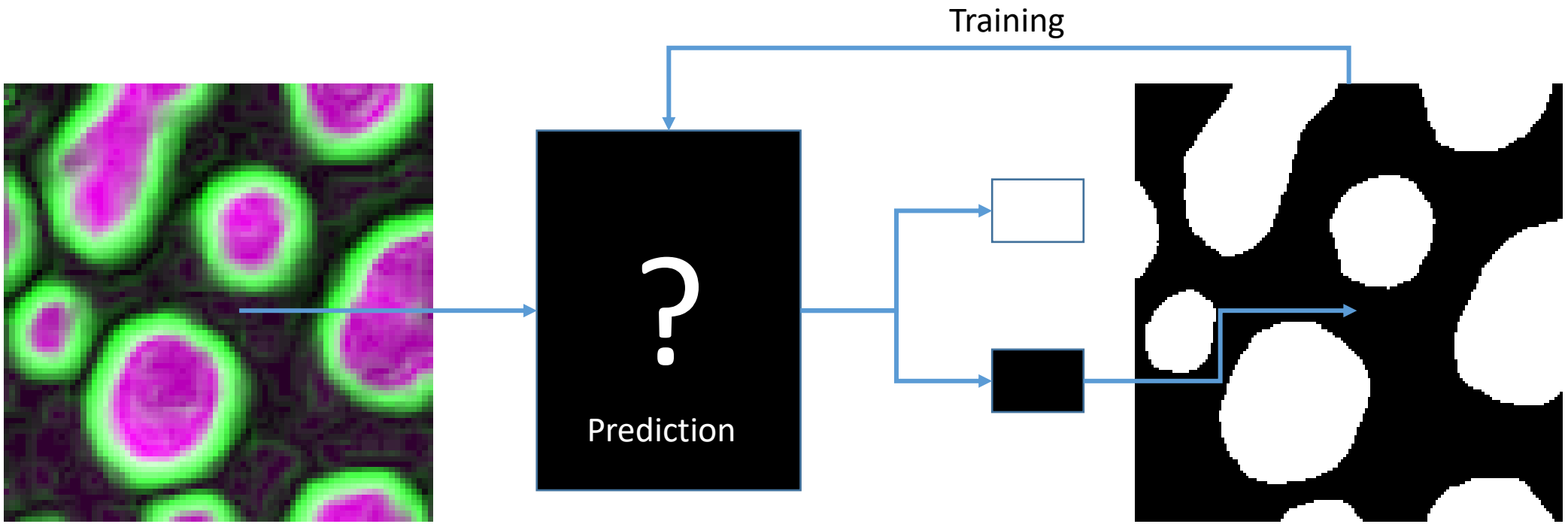
- The right approach depends on data, computational resources and desired quality



Adapted from https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Machine learning for image segmentation

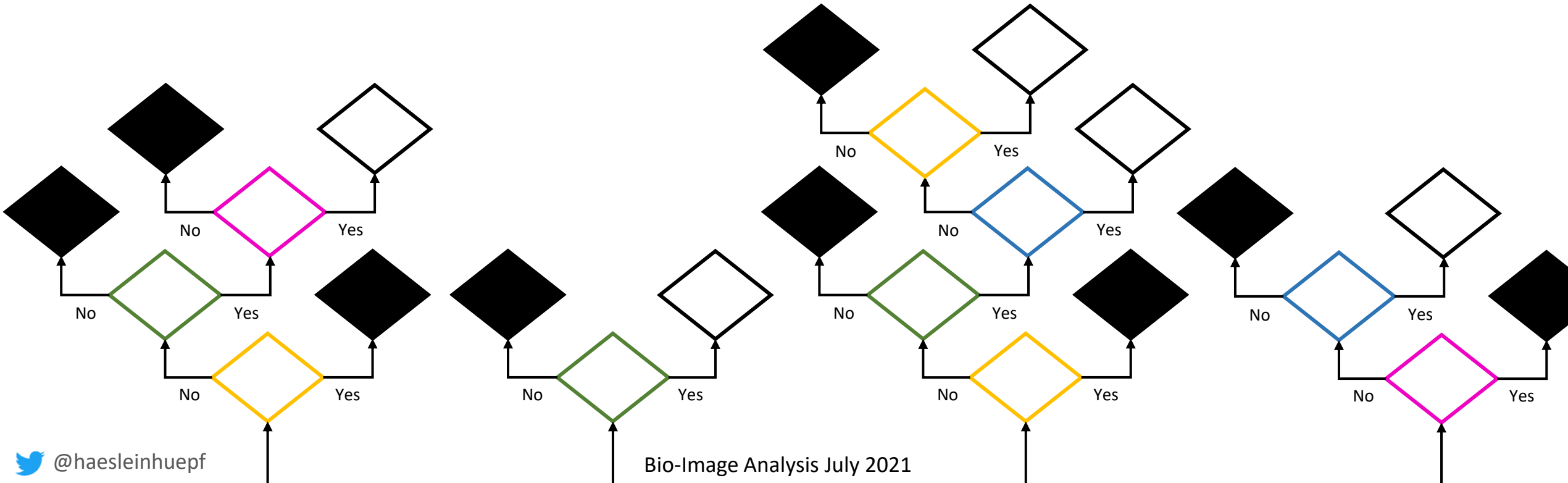
- *Supervised* machine learning: We give the computer some ground truth to learn from
- The computer derives a *model* or a *classifier* which can judge if a pixel should be foreground (white) or background (black)
- Example: Binary classifier



Random forest based image segmentation

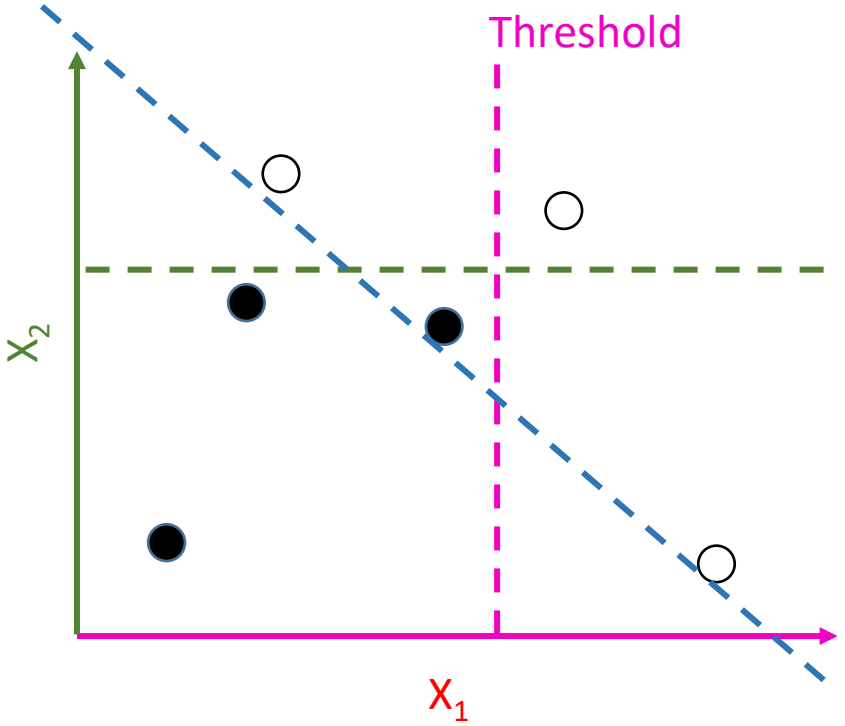
- Decision trees are classifiers, they decide if a pixel should be white or black
- Random decision trees are randomly initialized, afterwards evaluated and selected
- Random forests consist of many random decision trees

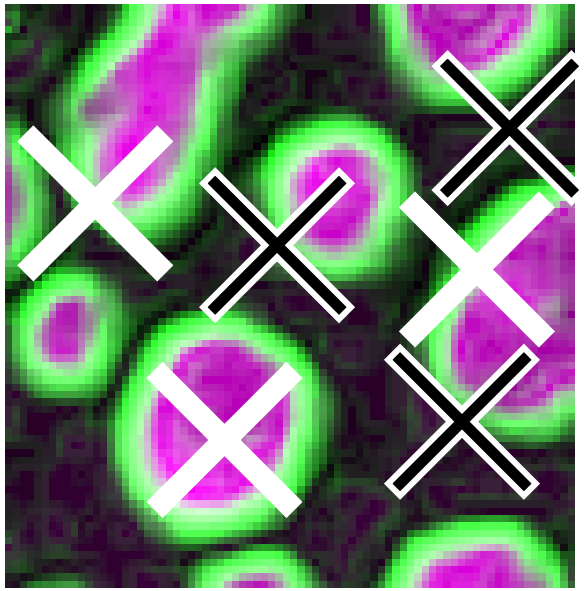
- Example: Random forest of binary decision trees

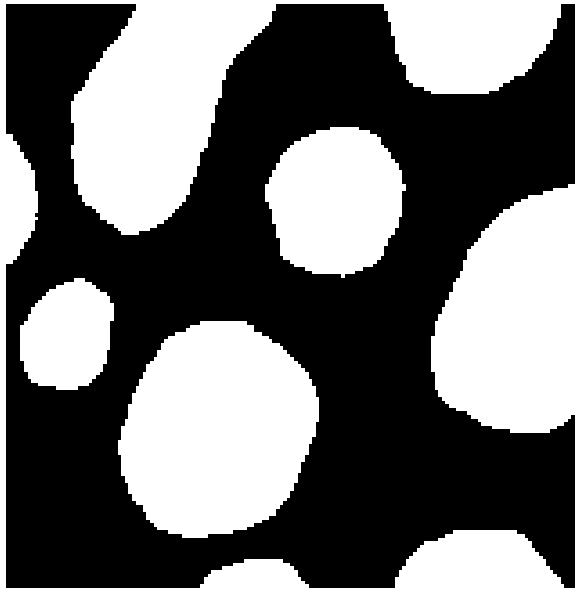


Deriving random decision trees

- For efficient processing, we randomly *sample* our data set
  - Individual pixels, their intensity and their classification



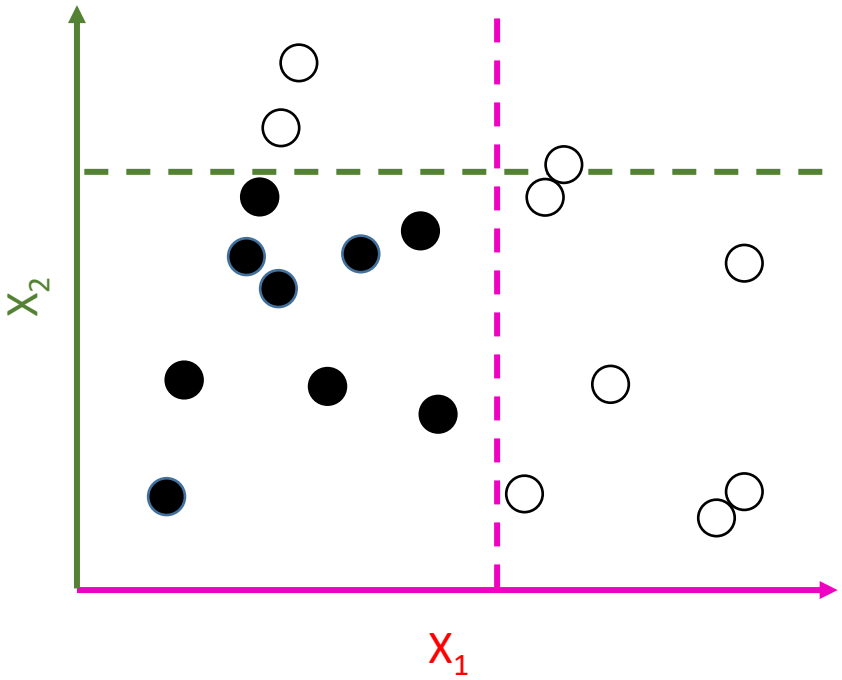


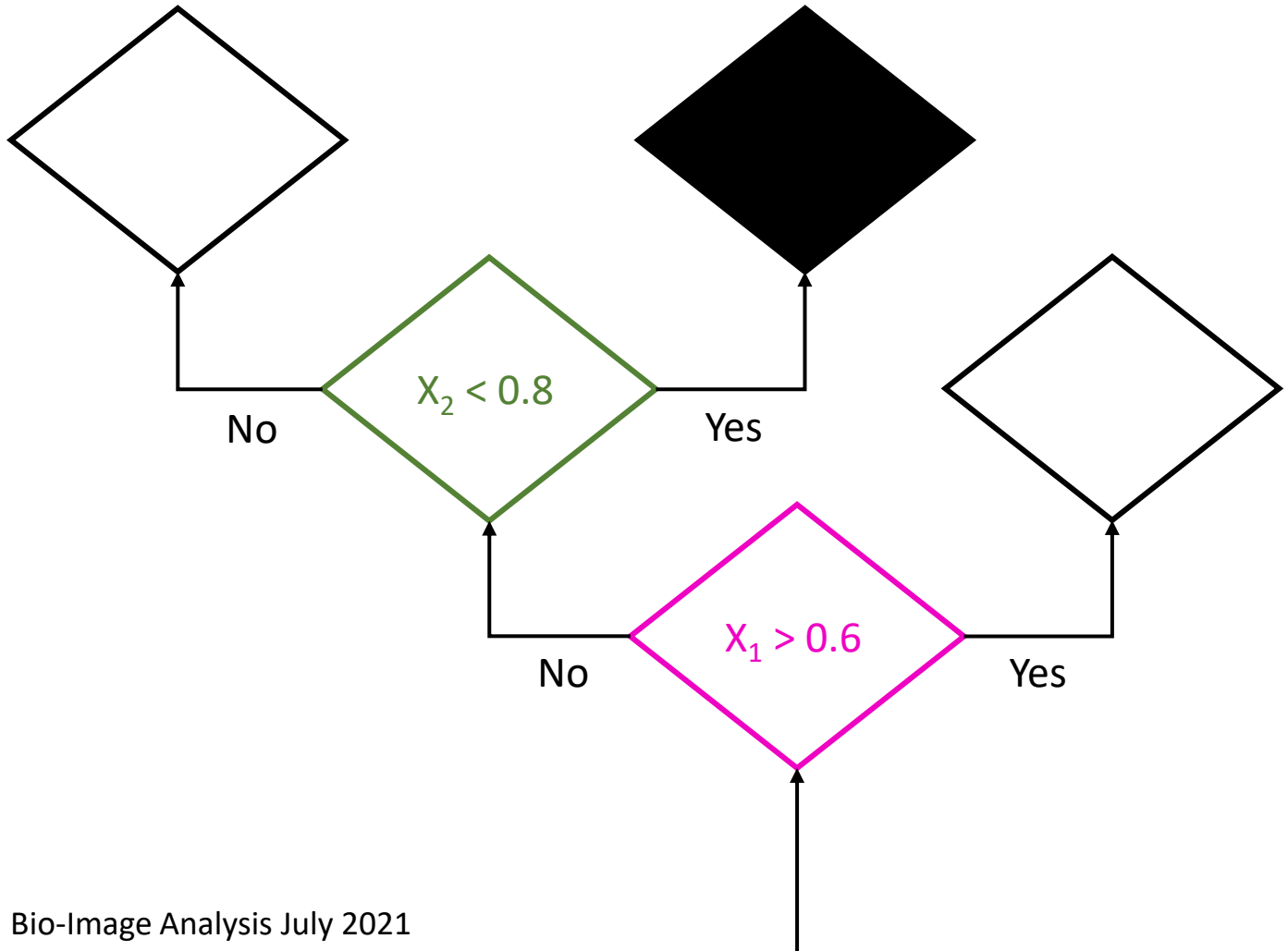


Note: You cannot use a single threshold to make the decision correctly

Deriving random decision trees

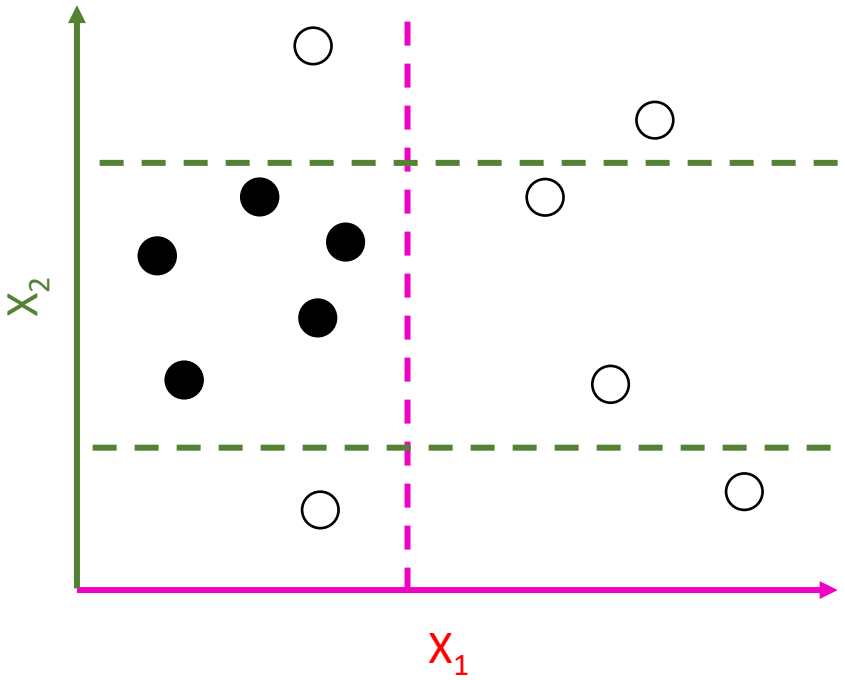
- Decision trees combine several thresholds on several parameters

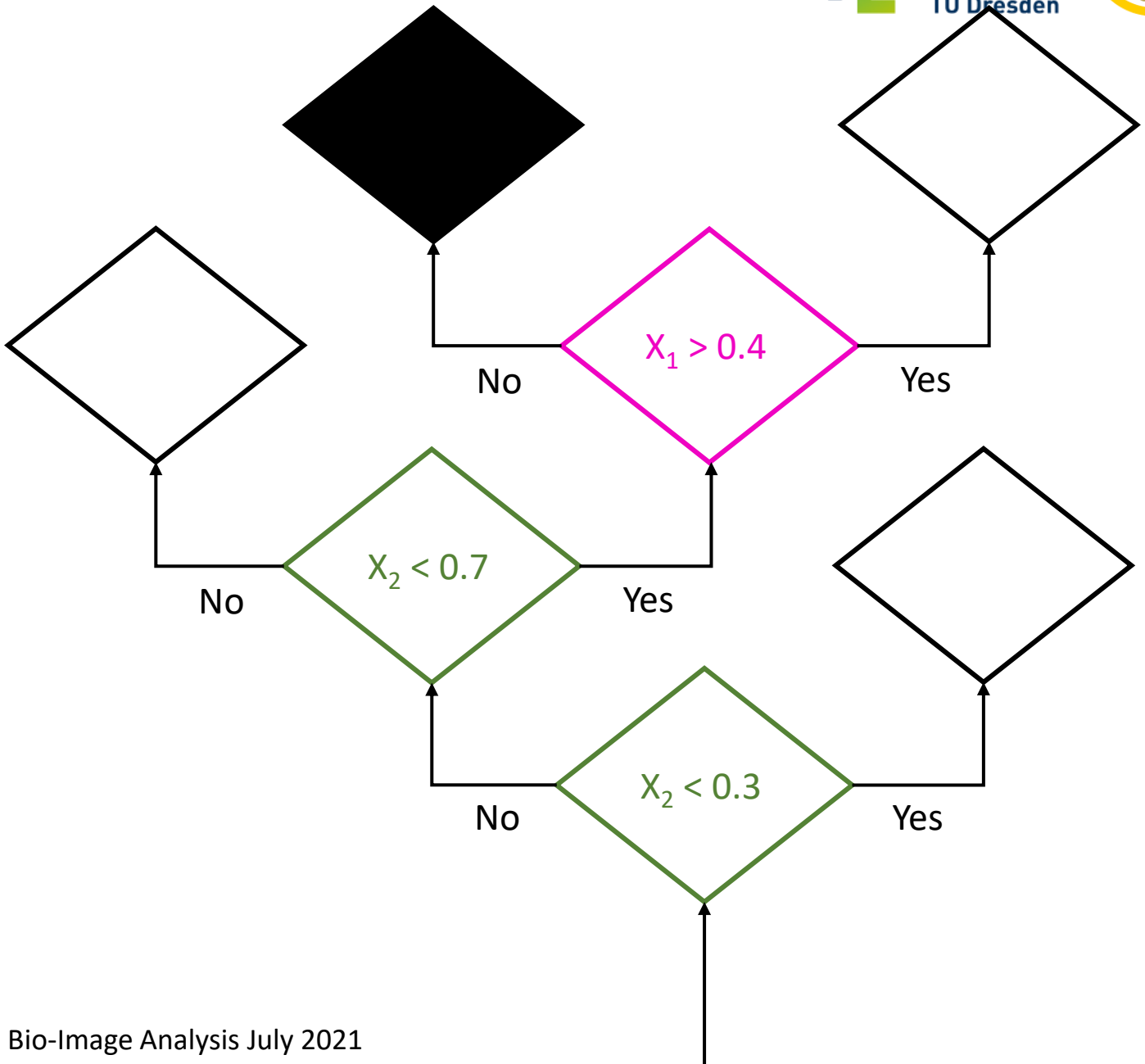




Deriving random decision trees

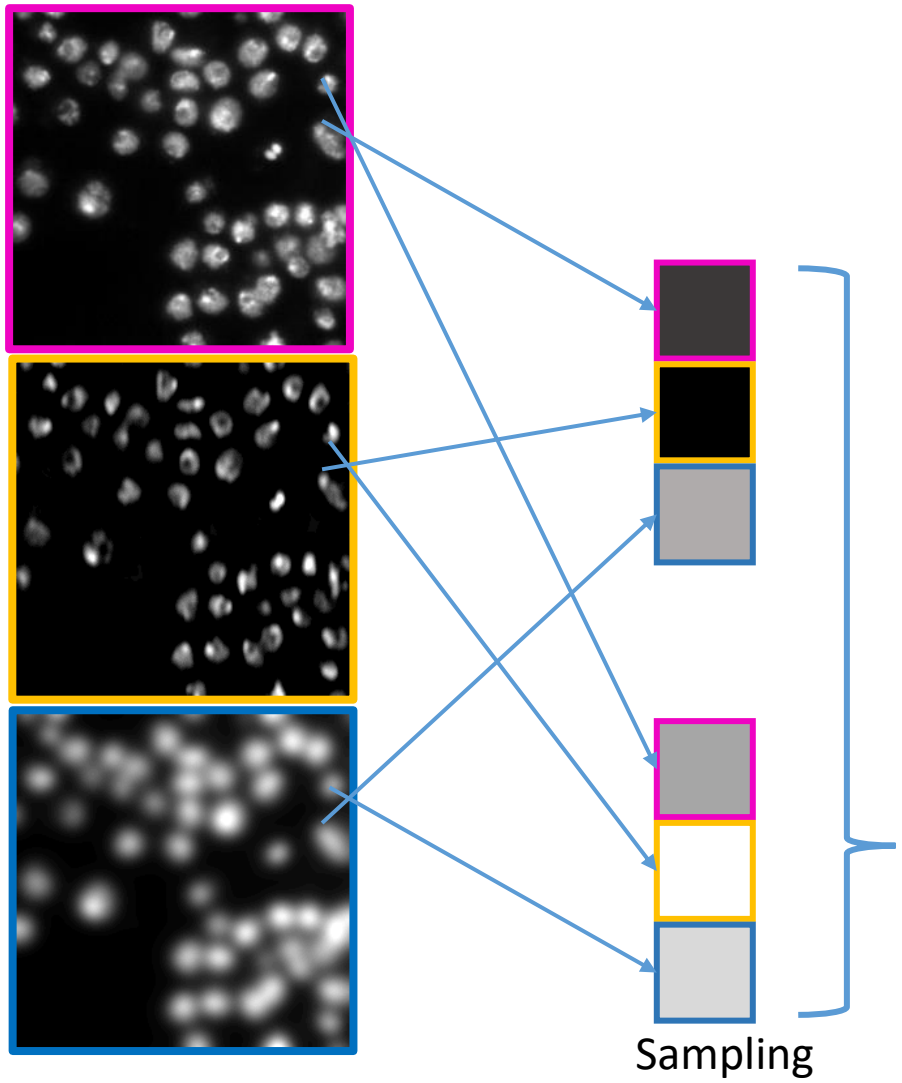
- Depending on sampling, the decision trees are different

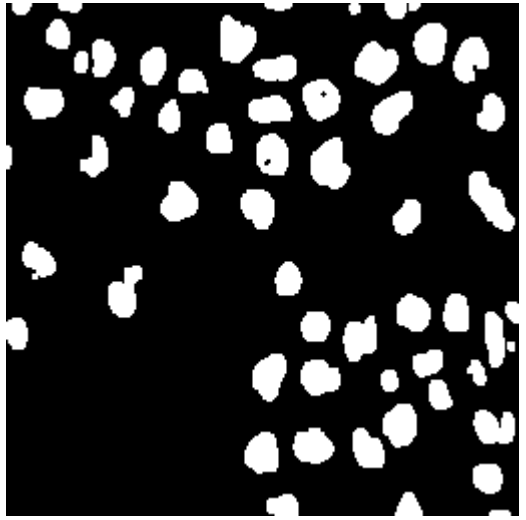




Random Forest Pixel Classifiers

- By comparing performance of individual decision trees, good ones can be selected, bad ones excluded.





Selection

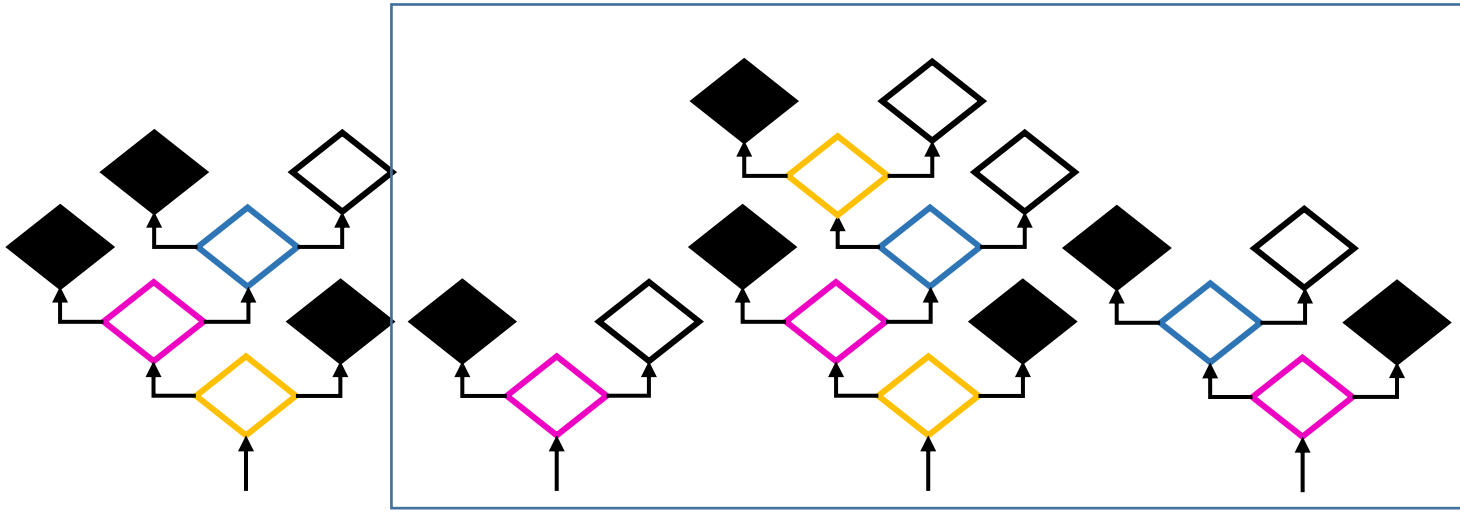
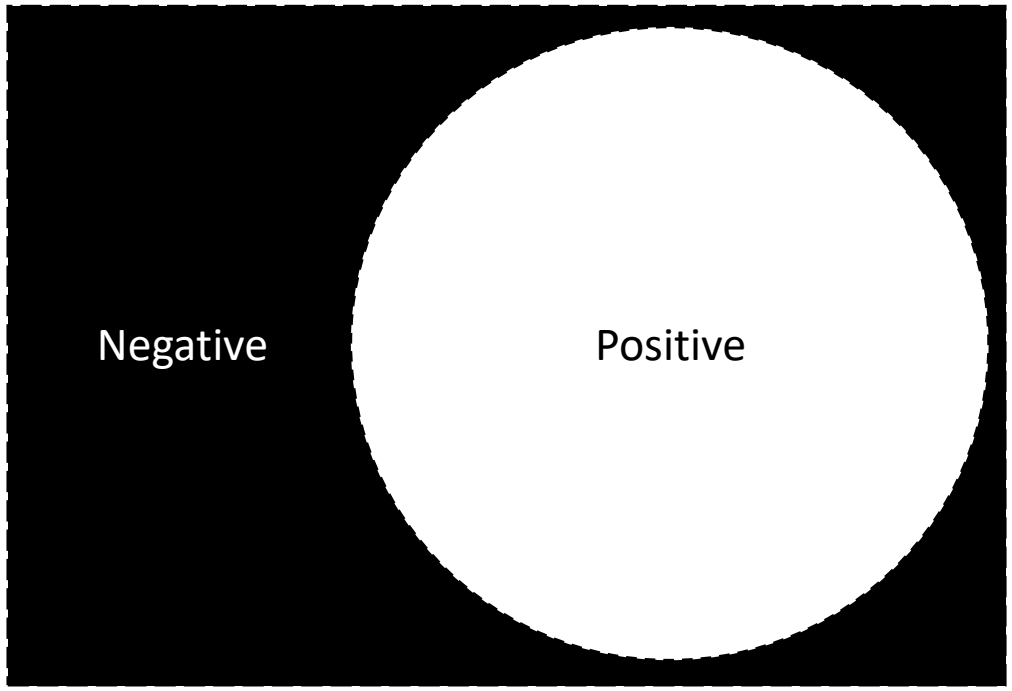


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

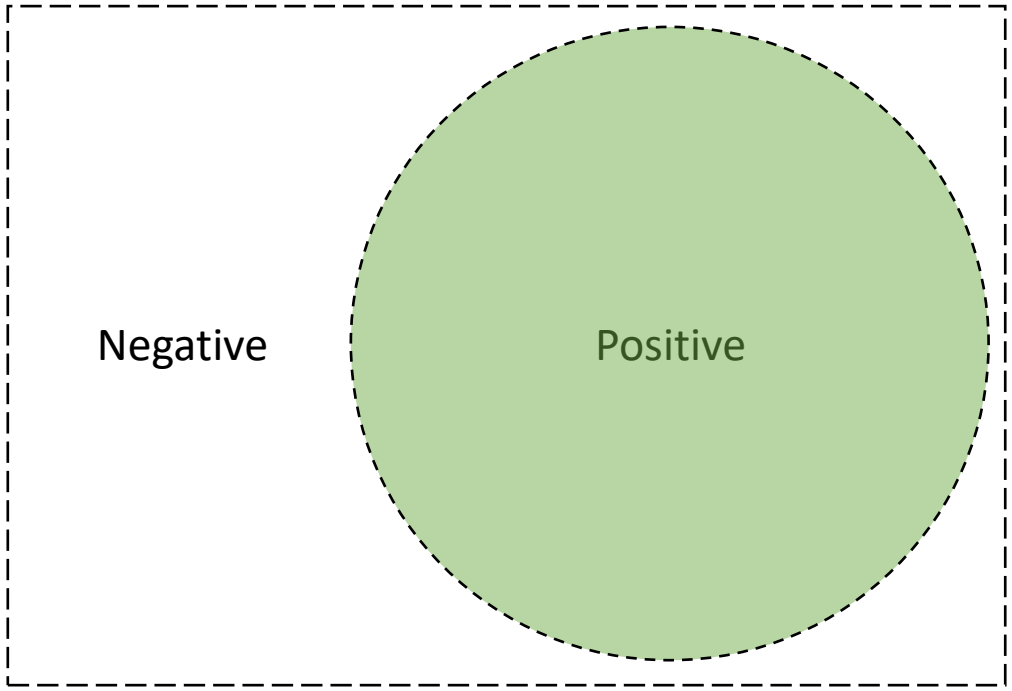
Algorithm evaluation

- In general
  - Define what's positive and what's negative.



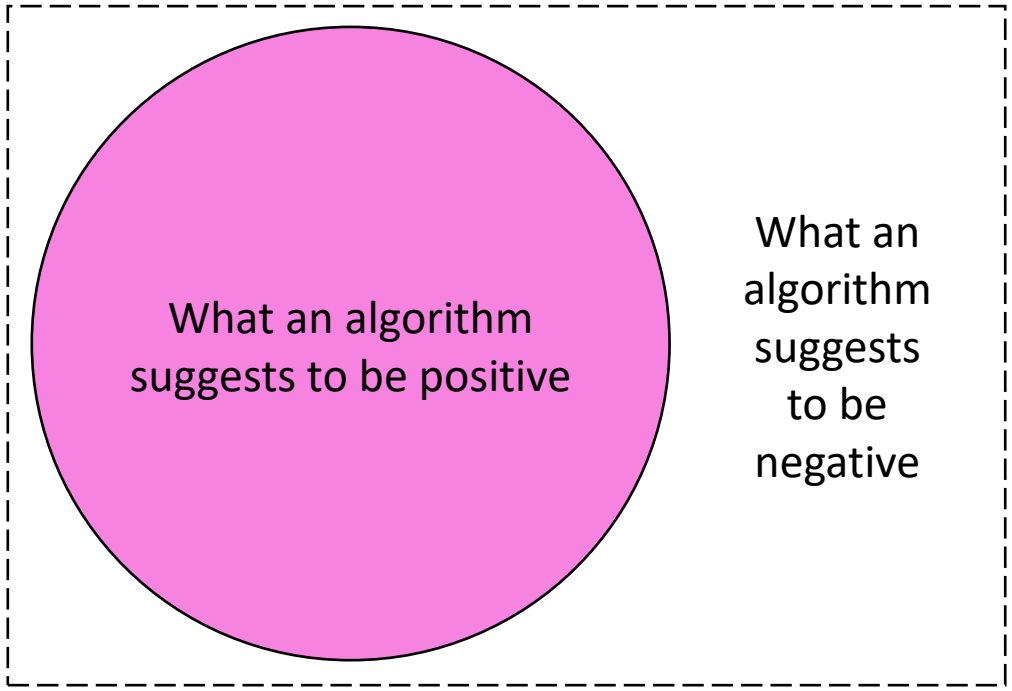
Algorithm evaluation

- In general
  - Define what's positive and what's negative.



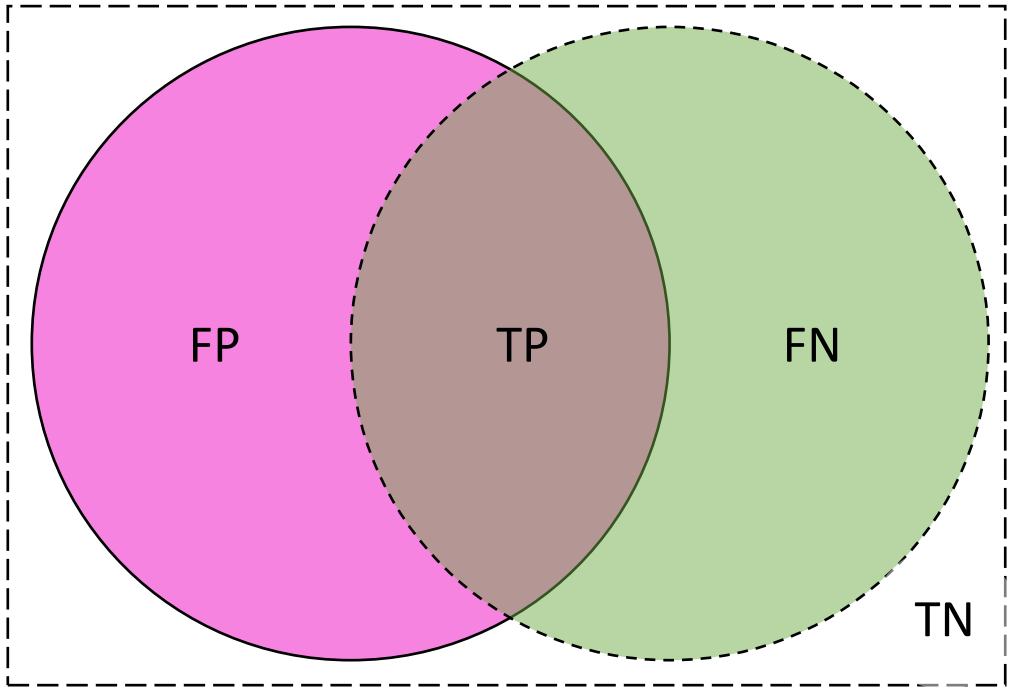
Algorithm evaluation

- In general
  - Define what's positive and what's negative.



Algorithm evaluation

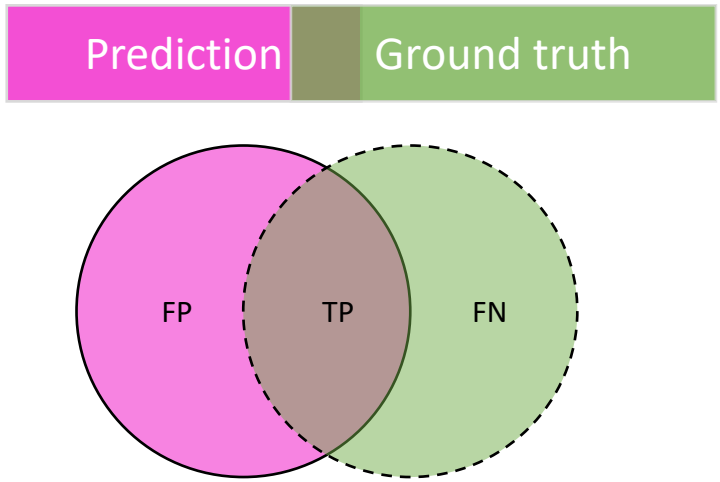
- In general
  - Define what's positive and what's negative.
  - Compare with a reference to figure out what was true and false
  - Welcome to the Theory of Sets



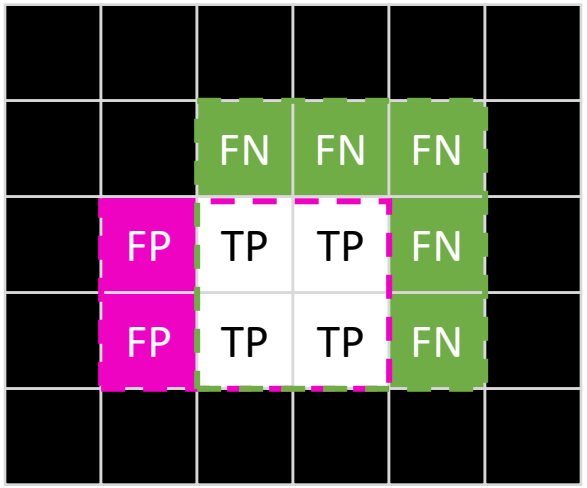
A	Prediction A
B	Reference B (ground truth)
ROI	Region of interest
TP	True-positive
FN	False-negative
FP	False-positive
TN	True-negative

Precision	$\frac{TP}{TP+FP}$	What fraction of points that were predicted as positives were really positive?
Recall (a.k.a. sensitivity)	$\frac{TP}{TP+FN}$	What fraction of positives points were predicted as positives?

Pixel-wise versus Object-wise evaluation



- Pixel wise: Segmentation quality



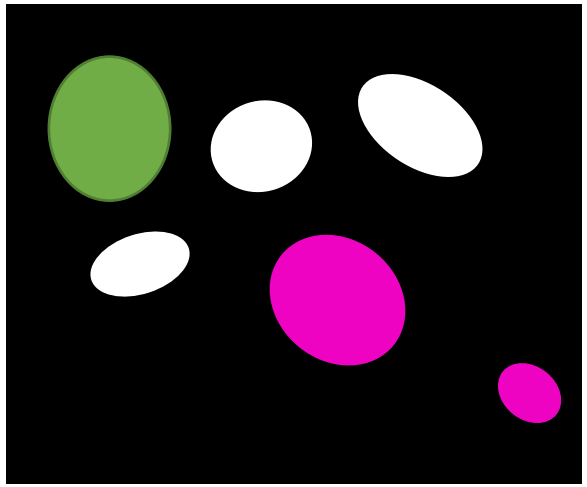
True-positive: 4

False-negative: 5

False-positive: 2

Precision: 4/6 = 66%
Recall: 4/9 = 44%

- Object wise: Detection quality



True-positive: 3

False-negative: 1

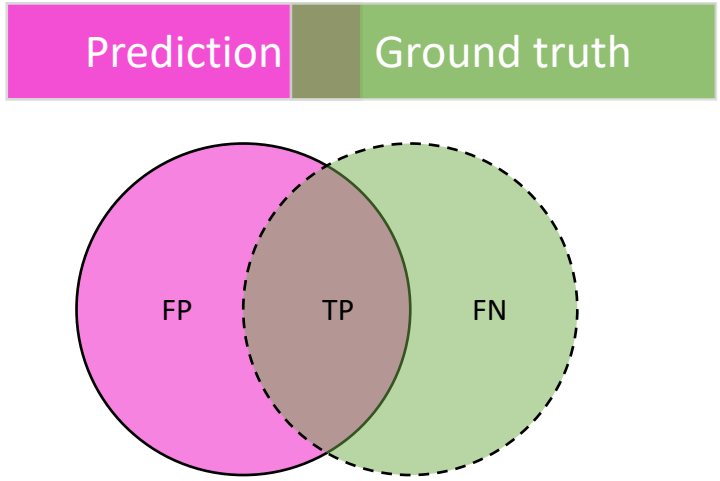
False-positive: 2

Precision: 3/4 = 75%
Recall: 4/9 = 44%

Precision $\frac{TP}{TP+FP}$

Recall (a.k.a. sensitivity) $\frac{TP}{TP+FN}$

Pixel-wise versus Object-wise evaluation



Overlap
(a.k.a. Jaccard index)

$$\frac{TP}{TP + FN + FP}$$

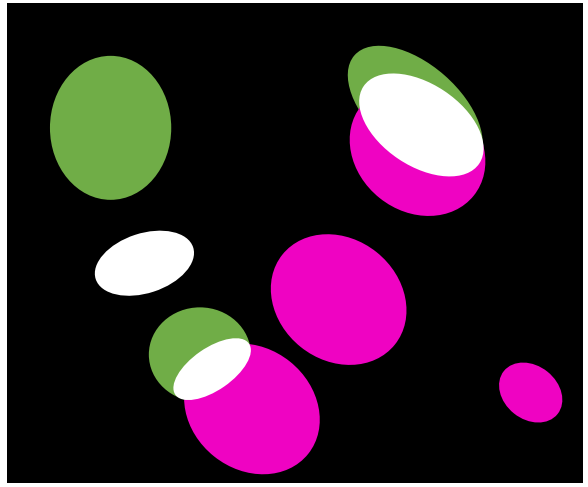
Precision

$$\frac{TP}{TP + FP}$$

Recall
(a.k.a. sensitivity)

$$\frac{TP}{TP + FN}$$

- In practice: Mixed



Objects with at least 50% *pixel-wise overlap* between P and GT
True positive: 2

False positives: 3

False negatives: 2

Precision: 2/5 = 40%
Recall: 2/4 = 50%

Algorithm evaluation

- Voxel-wise Youden-Index

$$YI = p_{TP} + p_{TN} - 1$$

- Volume error

$$\Delta_V = V_A - V_B$$

$$\delta_V = \frac{\Delta_V}{V_B}$$

- Dice Index

$$DI(A,B) = \frac{2\,|A \cap B|}{|A| + |B|}$$

- Jaccard Index

$$JI(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{DI}{2 - DI}$$

- Contour distance

$$d_{e,min}(a,B) = \min(d_e(a,b)|b \in B)$$

$$\bar{d}_c(A,B) = \frac{\sum_{\forall a \in C(A)} d_{e,min}(a, C(B))}{|C(A)|}$$

$$\bar{d}_{bil,c}(A,B) = \frac{\bar{d}_c(A,B) + \bar{d}_c(B,A)}{2}$$

- Hausdorff distance

$$d_H(A,B) = \max(d_{e,min}(a,B)|a \in A)$$

$$d_{bil,H}(A,B) = \max(d_H(A,B), d_H(B,A))$$

- Simplified Hausdorff distance

$$d_H(A,B) = \max(d_{e,min}(a, C(B))|a \in C(A))$$

- Volume standard deviation

$$\delta_{\bar{V}} = 2\frac{|V_A - V_B|}{|V_A + V_B|}$$

- Classification error

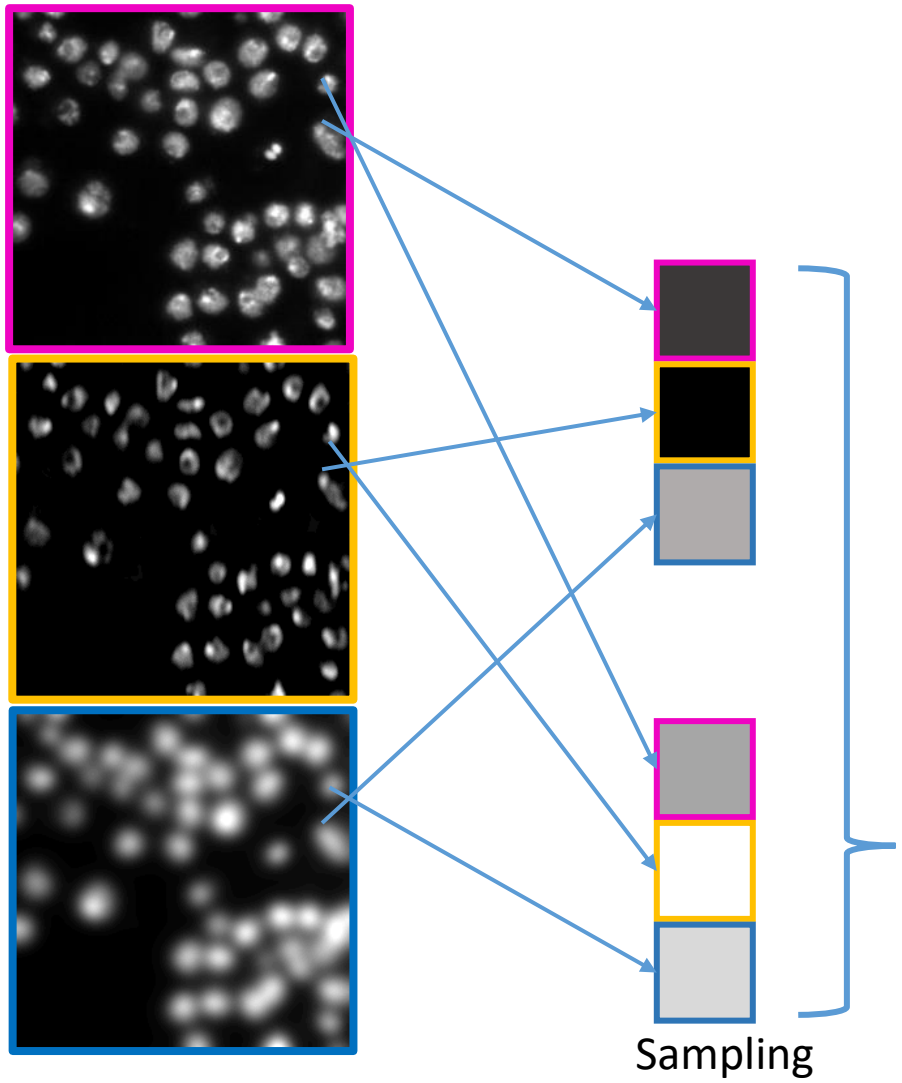
$$e_{Class} = \frac{H}{|TP| + |FN|}$$

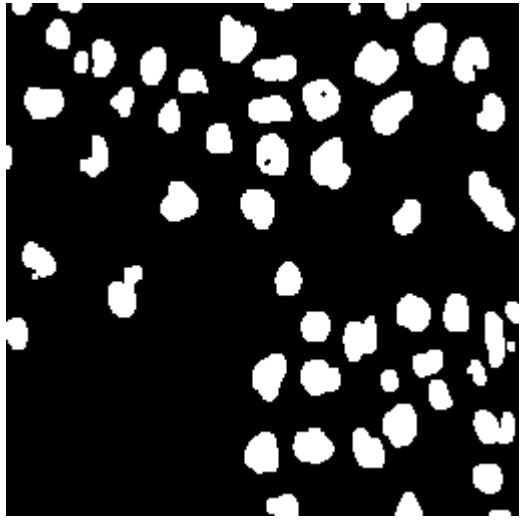
- Hamming distance

$$d_h = |A \cup B| - |A \cap B|$$
$$= |FP| + |FN|$$

Random Forest Pixel Classifiers

- By comparing performance of individual decision trees, good ones can be selected, bad ones excluded.





Selection

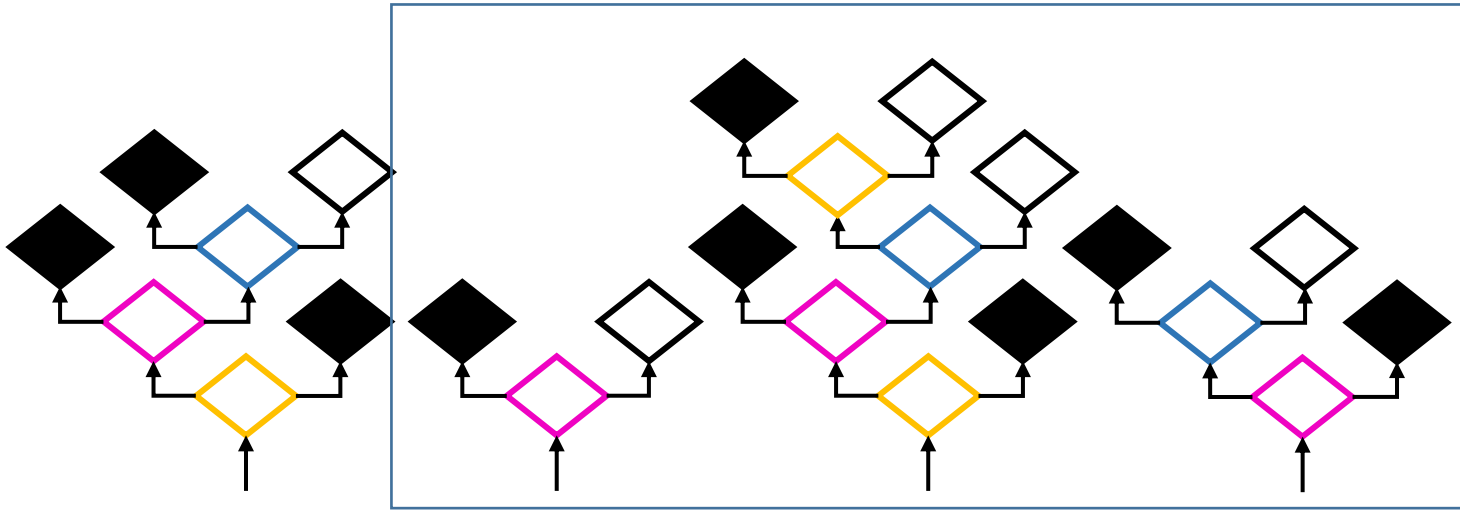
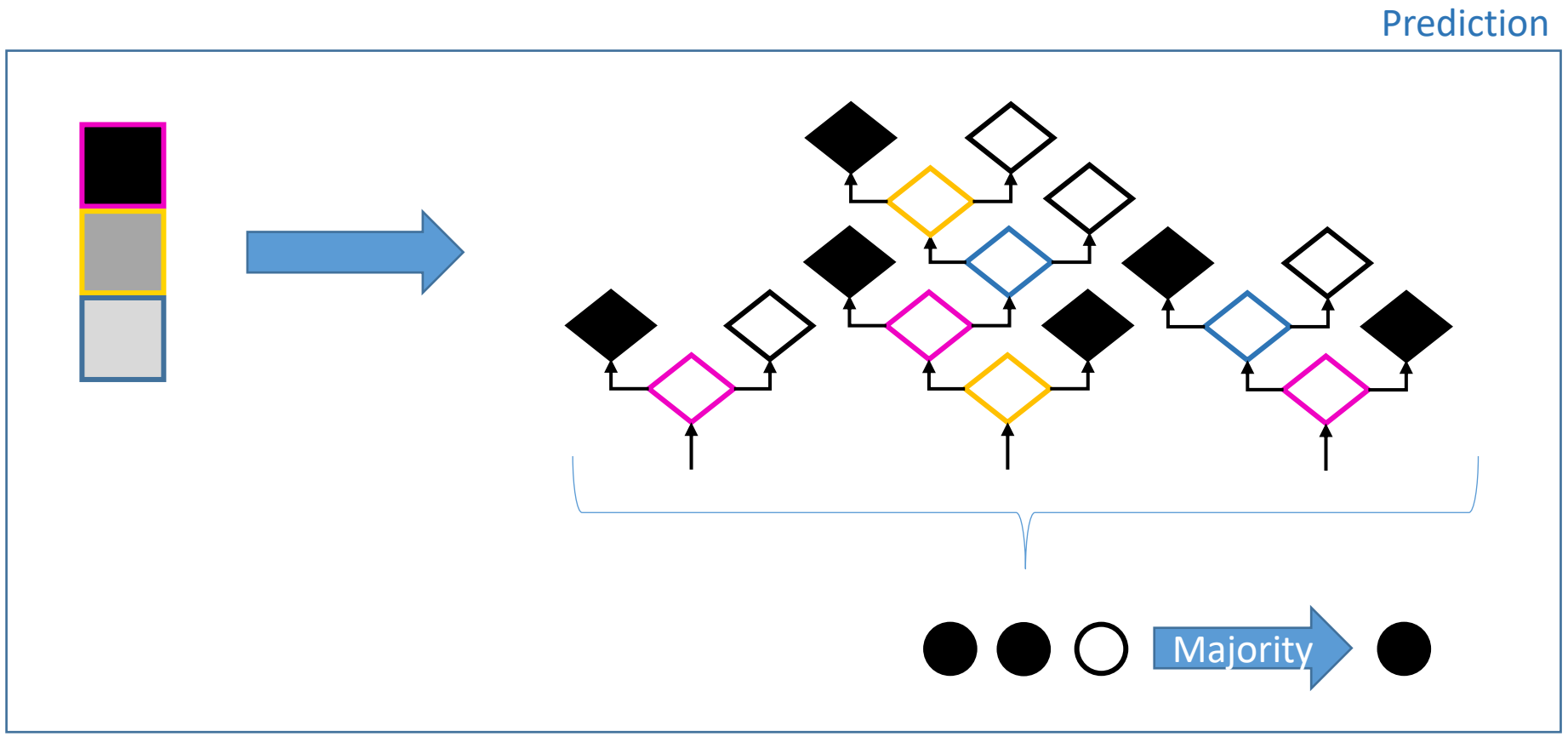


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Random Forest Pixel Classifiers

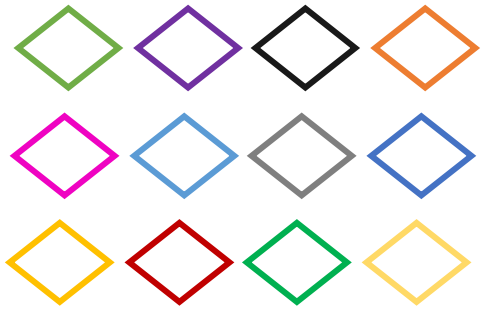
- Combination of individual tree decisions by voting or max / mean



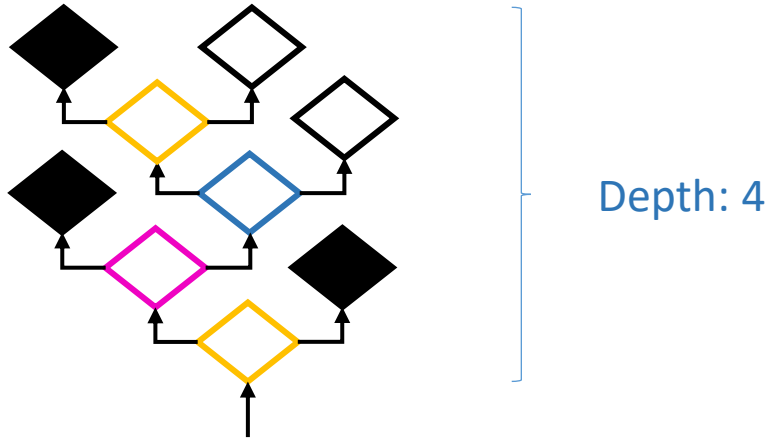
Random Forest Pixel Classifiers

- Typical numbers for pixel classifiers in microscopy

Available features: > 20

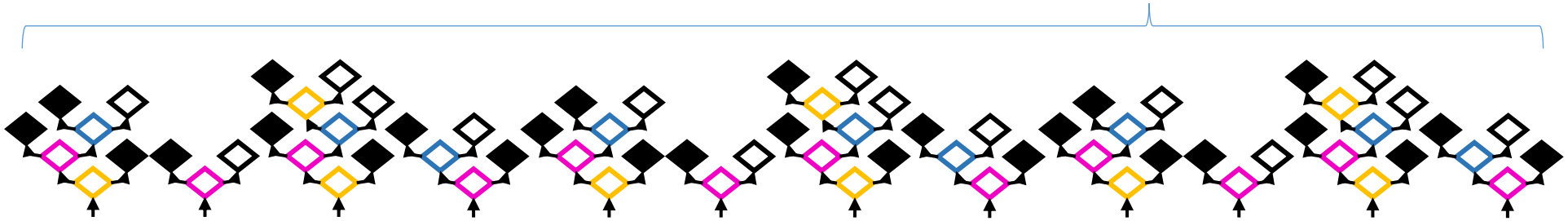


- Gaussian blur image
- DoG image
- LoG image
- Hessian
- ….



Selected features: <= depth

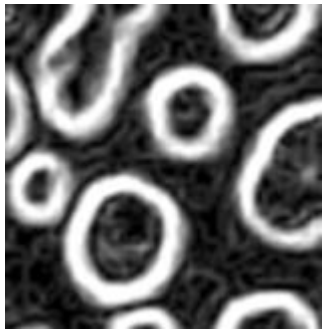
Number of trees: > 100

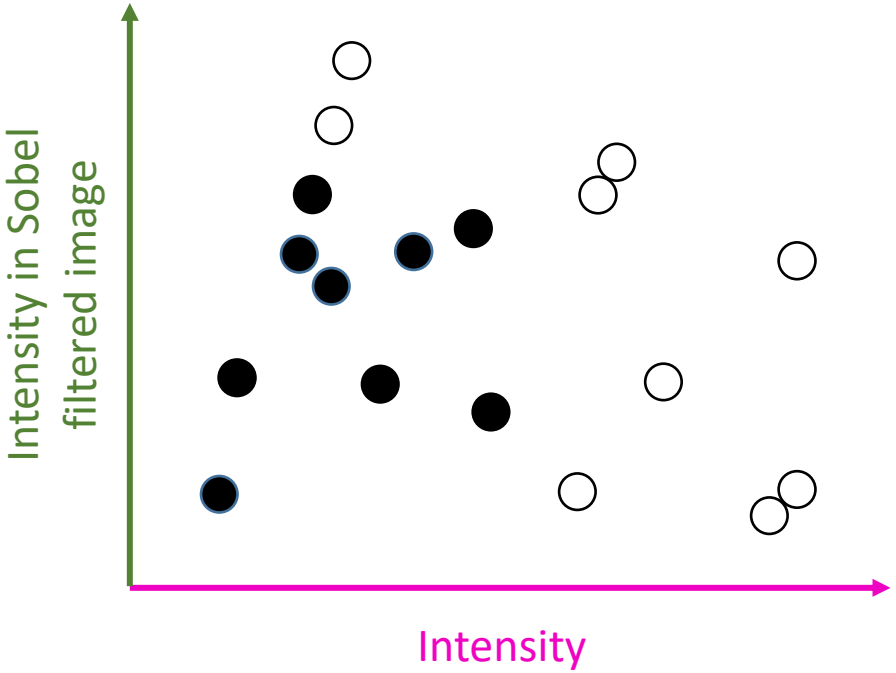


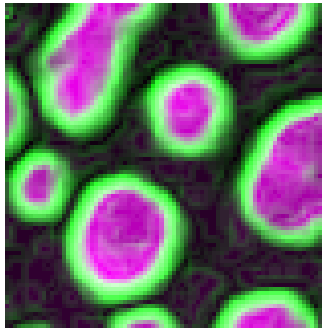
Object classification

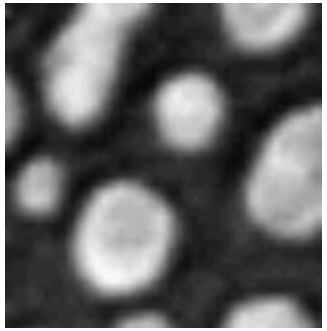
- What if we exchange pixel features with object features?

Pixel classification

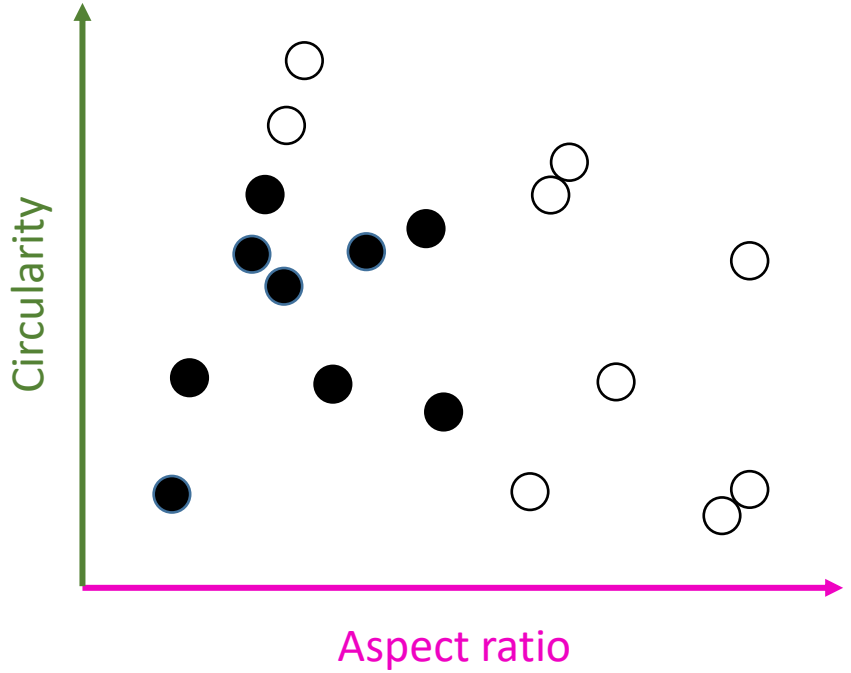








Object classification



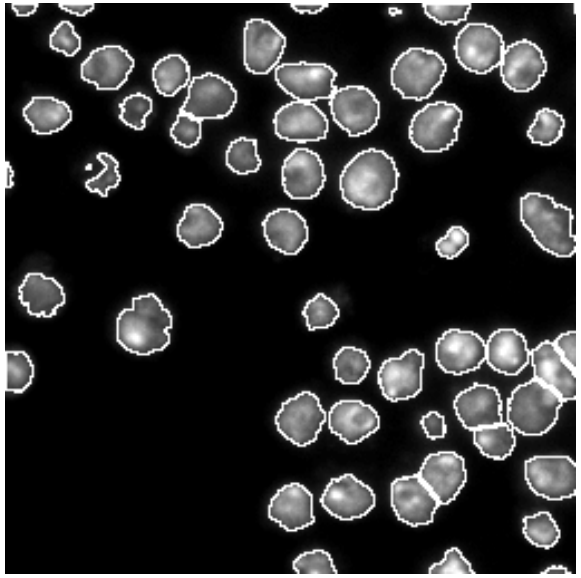
- The algorithms work the same but with different
  - Features
  - Number of features
  - Tree / forest parameters
  - Selection criteria

Object classification

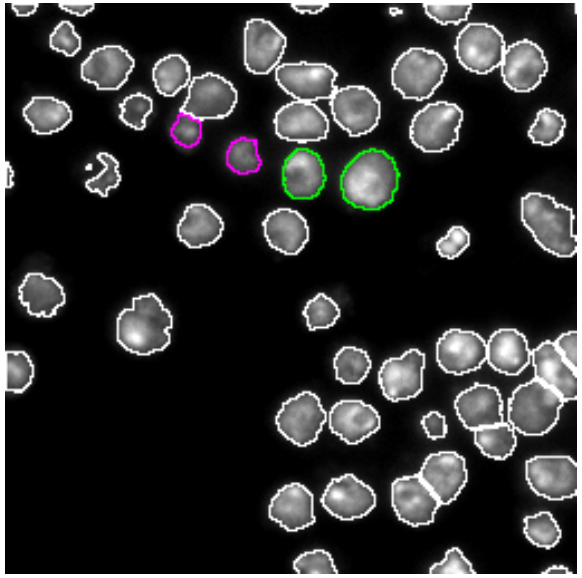
- Typical use cases
  - Shape differentiation
  - Cell-cycle differentiation
  - Eliminate wrong / bad segmentations

Warning: If you exclude nuclei depending on their shape, you're not allowed to analyse shape further.

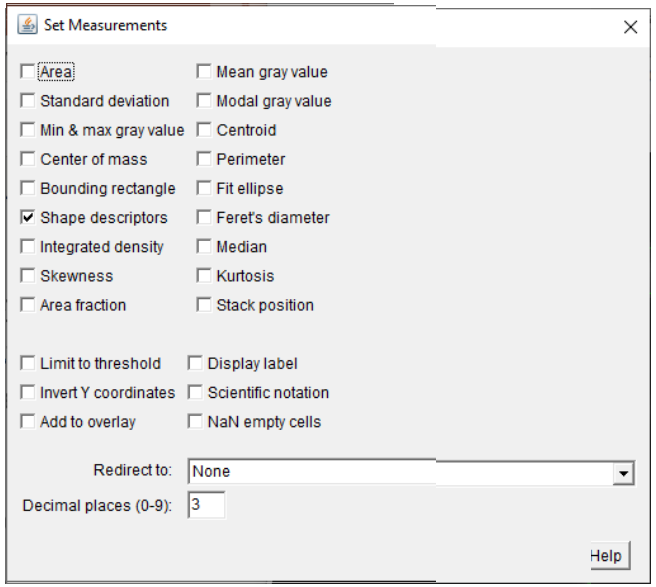
Objects



Ground truth annotation



Feature selection



Prediction

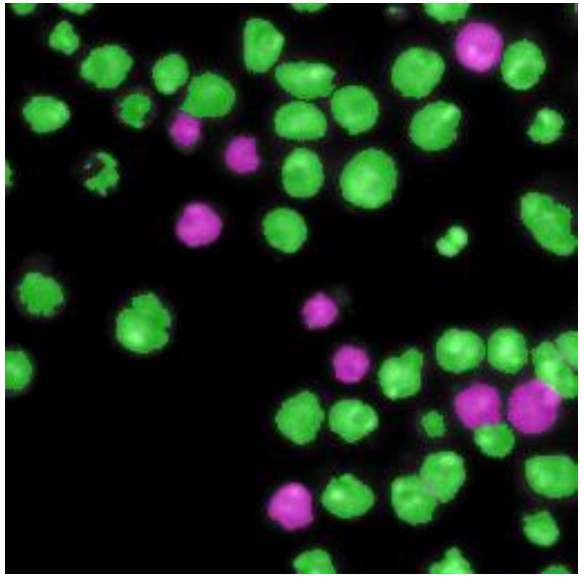
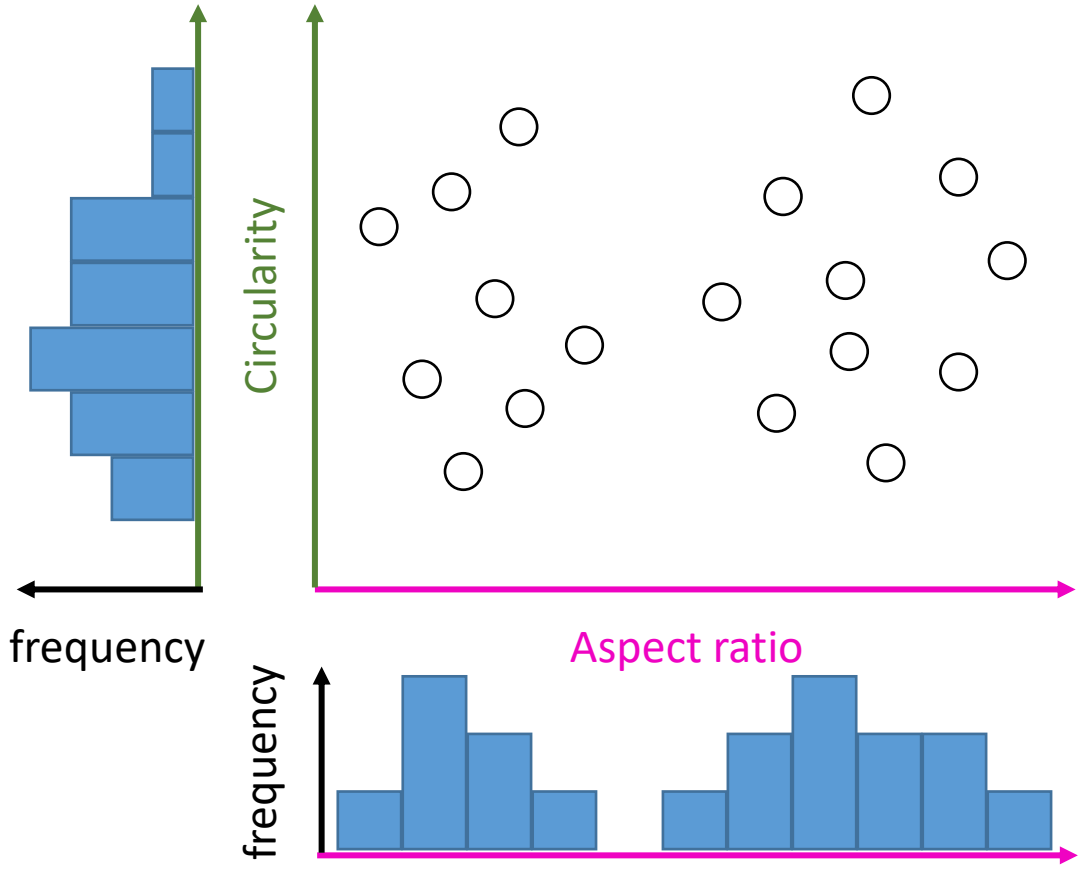


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

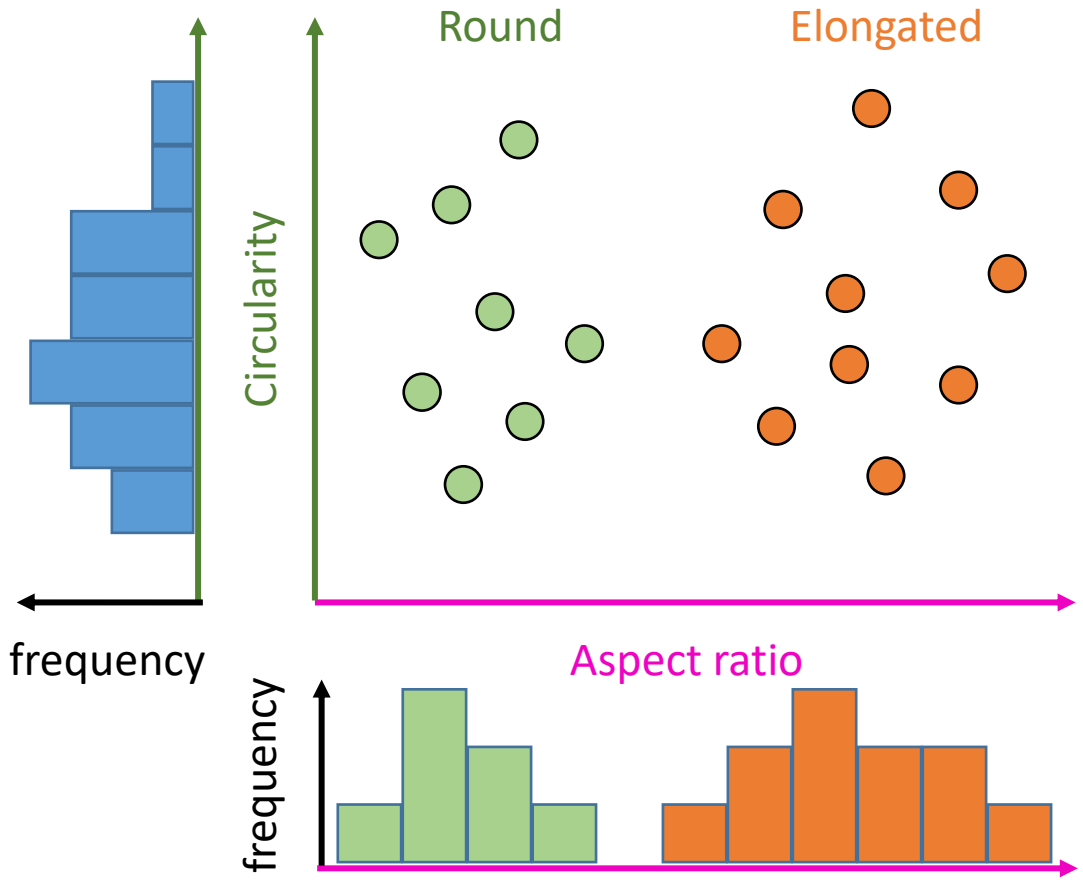
Unsupervised machine learning

- If you don't provide ground truth, the algorithm is *unsupervised*.



Unsupervised machine learning

- If you don't provide ground truth, the algorithm is unsupervised.
- Nevertheless, algorithms can tell us something about the data



Further reading:
- Principal component analysis
- Cluster analysis

Machine learning for image analysis

- In machine learning, we typically select features for training our classifier

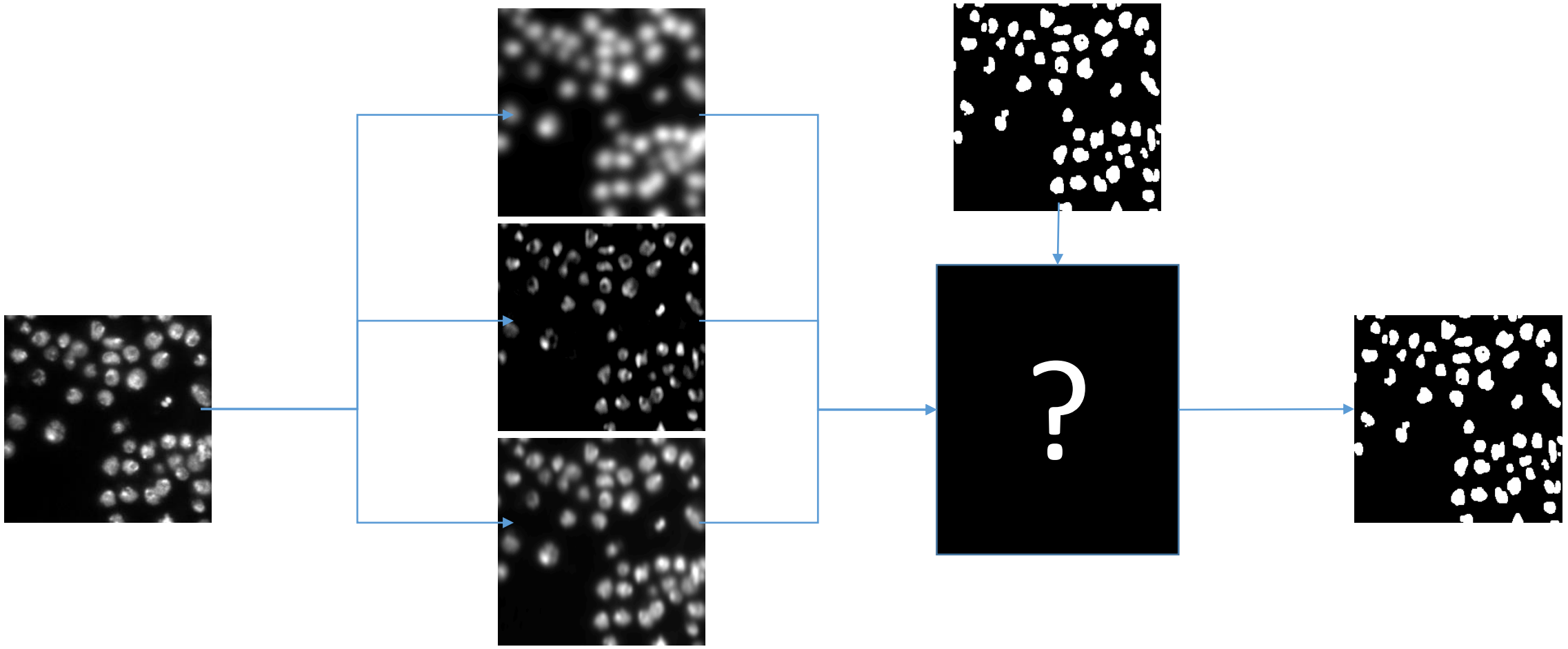
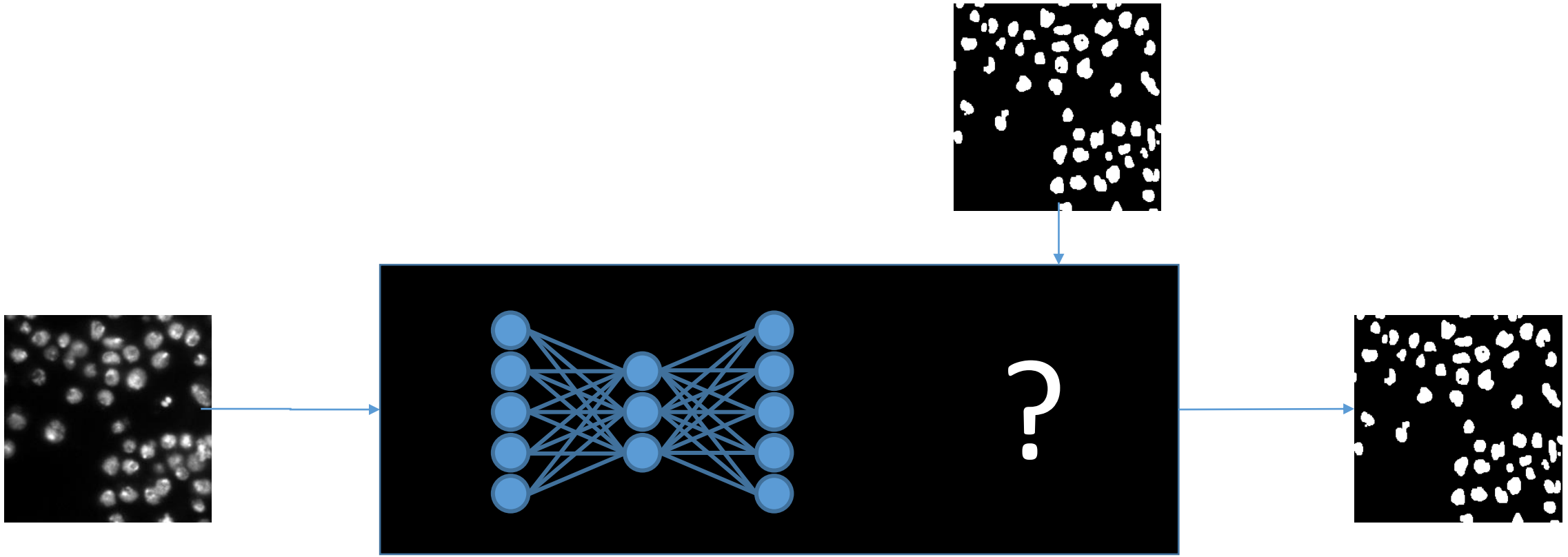


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Deep learning for image analysis

- In deep learning, this selection becomes part of the black box



Convolutional neural networks

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

Machine Learning for Pixel Classification in Fiji

Robert Haase

With Material from

Ignacio Arganda-Carreras, Universidad del Pais Vasco

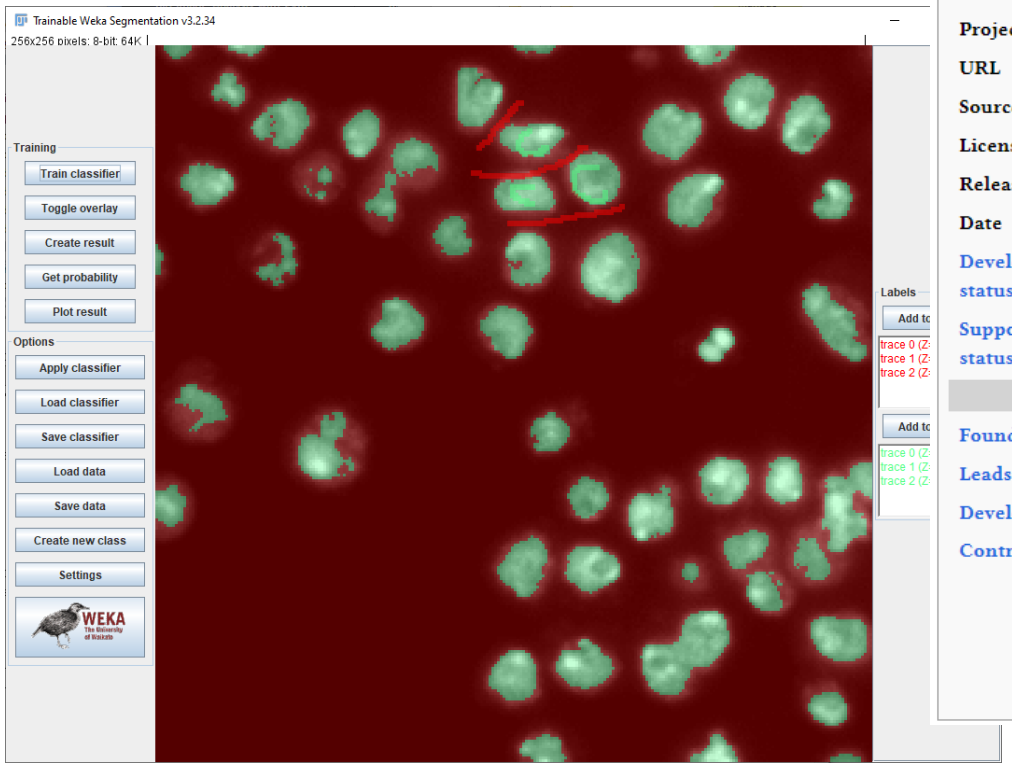
Matthias Arzt, Jug/Myers labs MPI CBG

@haesleinhuepf

July 2021

Trainable Weka Segmentation

- Fiji plugin for machine learning based image segmentation / pixel classification
- **Waikato Environment for Knowledge Analysis** (**Weka**), developed at the University of Waikato, New Zealand
- http://imagej.net/Trainable_Weka_Segmentation

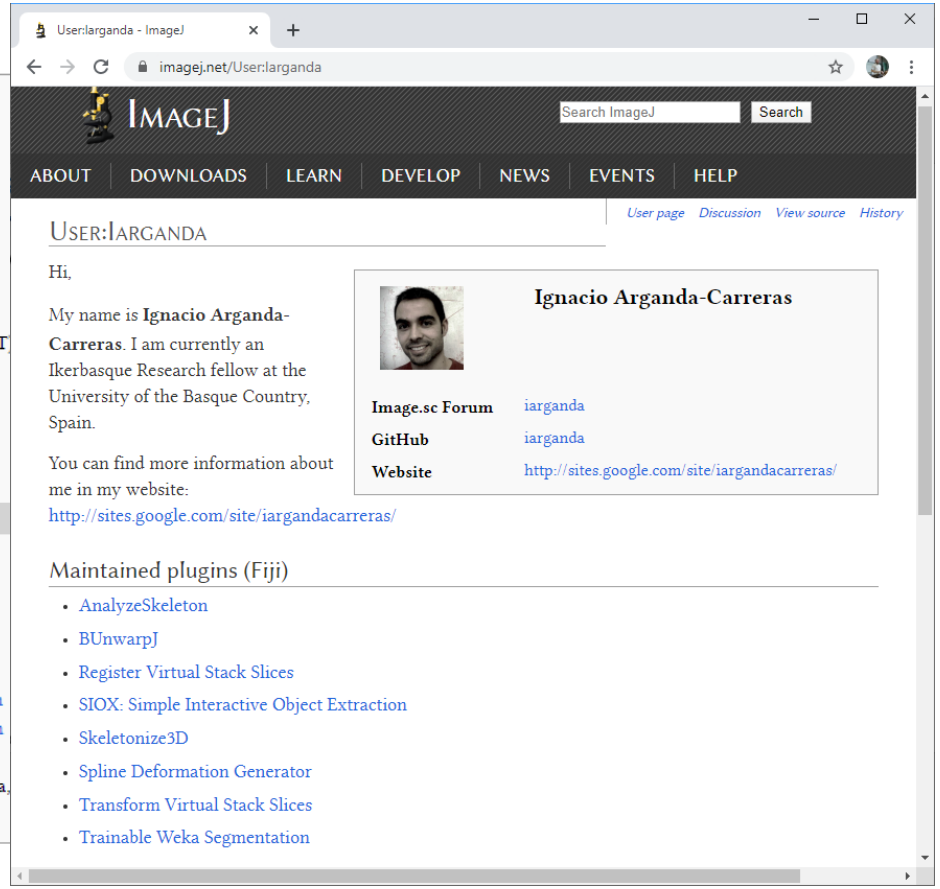


Trainable Segmentation

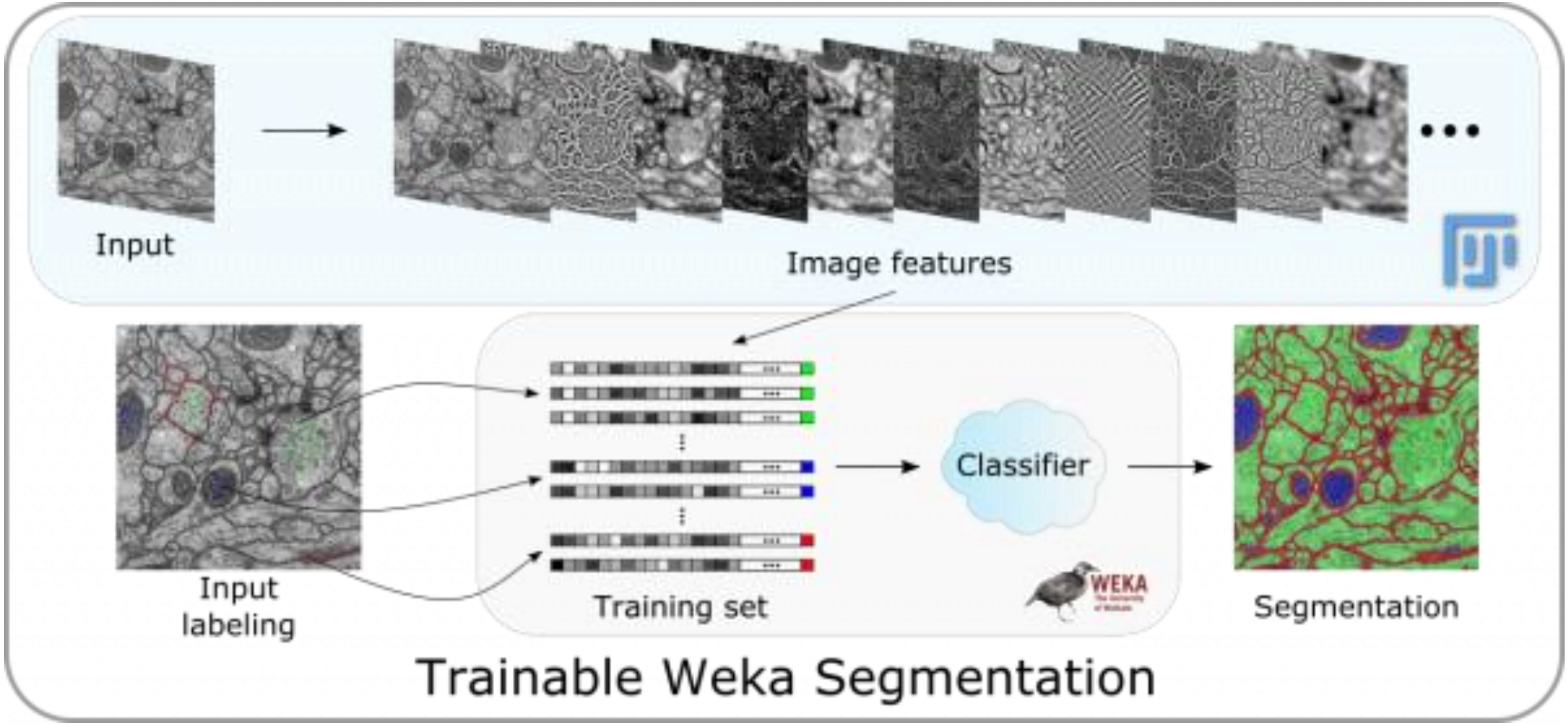
Project	Fiji
URL	https://imagej.net/Trainable_Segmentation
Source	on GitHub
License	GPLv3
Release	3.2.33
Date	Mon May 13 2019 12:01:00 GMT+0100 (CEST
Development status	Active
Support status	Active

Team

Founders	Verena Kaynig, Johannes Schindelin
Leads	Ignacio Arganda-Carreras
Developers	Ignacio Arganda-Carreras
Contributors	Verena Kaynig, Johannes Schindelin, Albert Cardona, Jan Eglinger, Patrice Freydiere, Jan Funke, Mark Hiner, Larry Lindsey, Christian Tischer, Eibe Frank, Richard Kirkby, Julien Prados, Fran Supek, Len Trigg, Santi Villalba, Yong Wang



Trainable Weka Segmentation



Source: http://imagej.net/Trainable_Weka_Segmentation

Random Forest Pixel Classifiers

- In Practice

- Plugins › Segmentation › Trainable Weka Segmentation

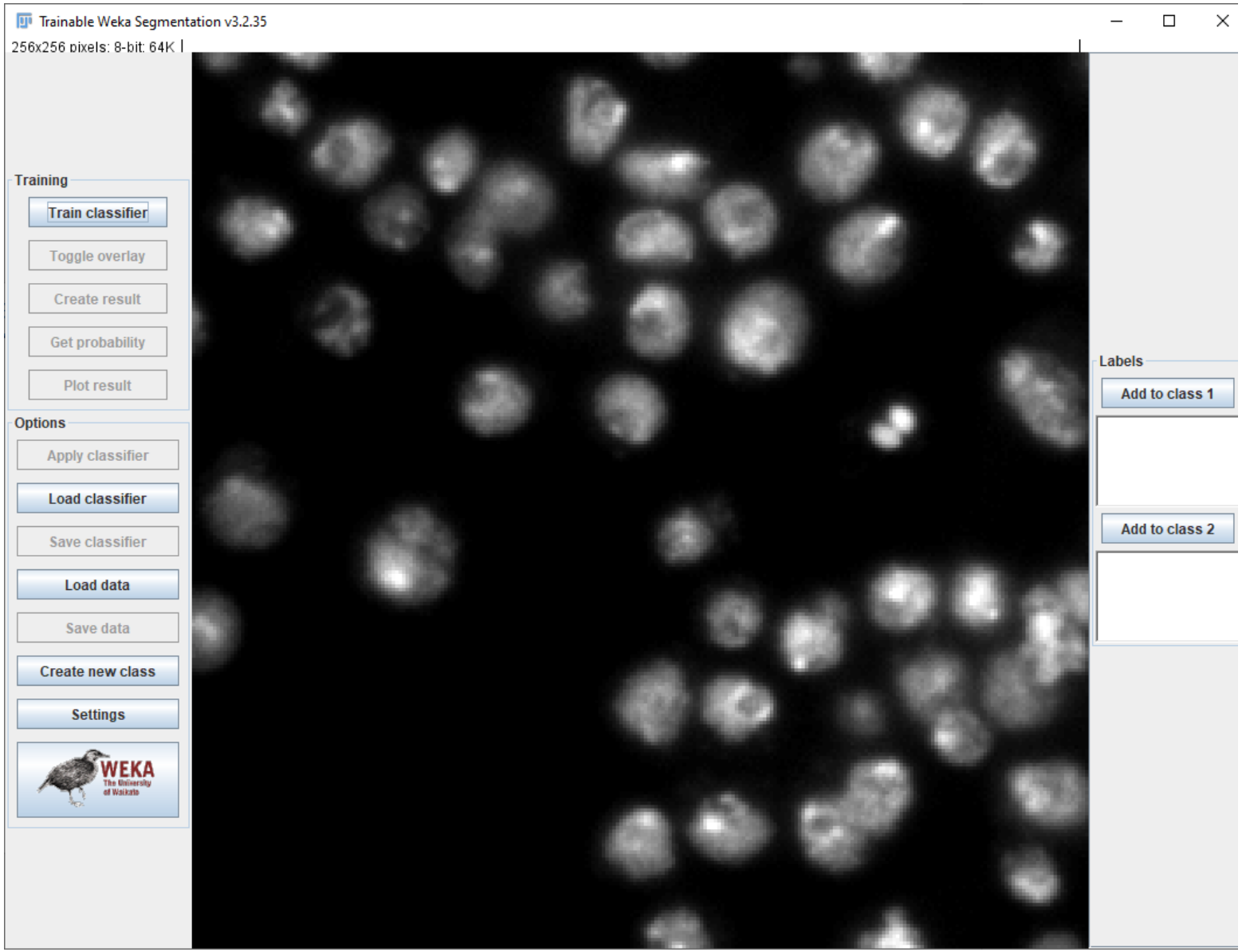


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Random Forest Pixel Classifiers

- In Practice

- Manually classify some pixels by drawing lines and adding them to class 1 or class 2

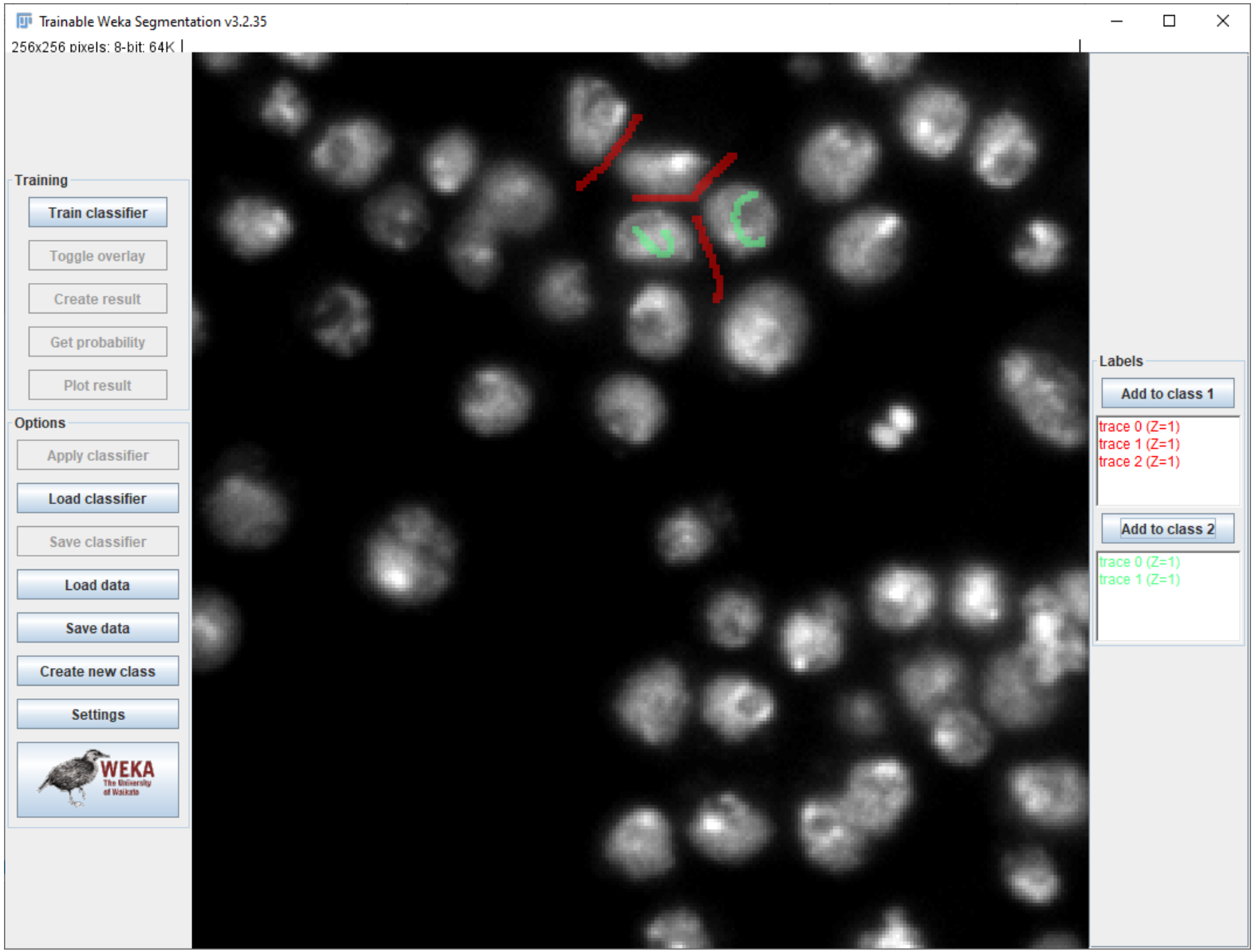


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Random Forest Pixel Classifiers

- In Practice

- Train a classifier!

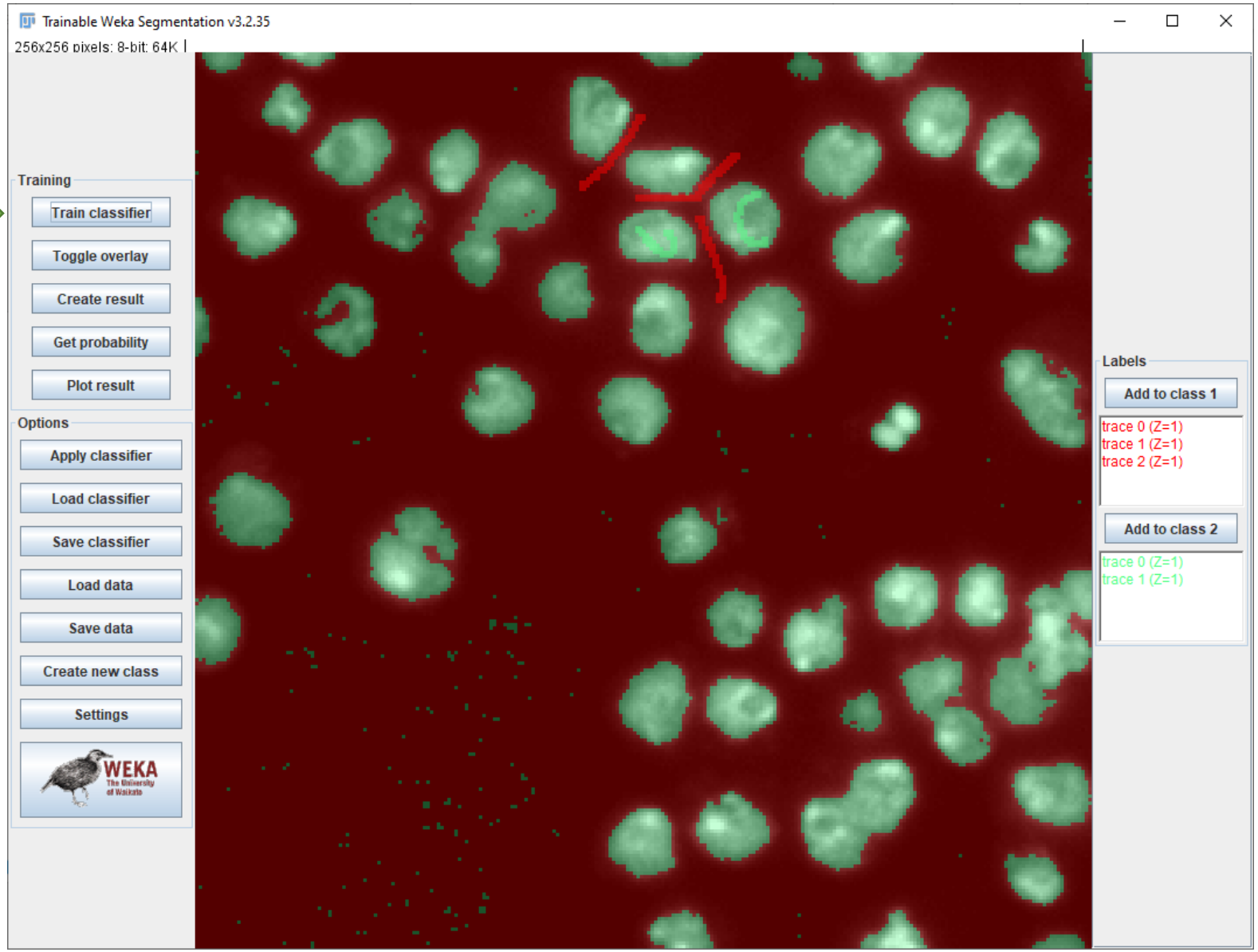


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Random Forest Pixel Classifiers

- In Practice

- Refine manual annotation to guide the training

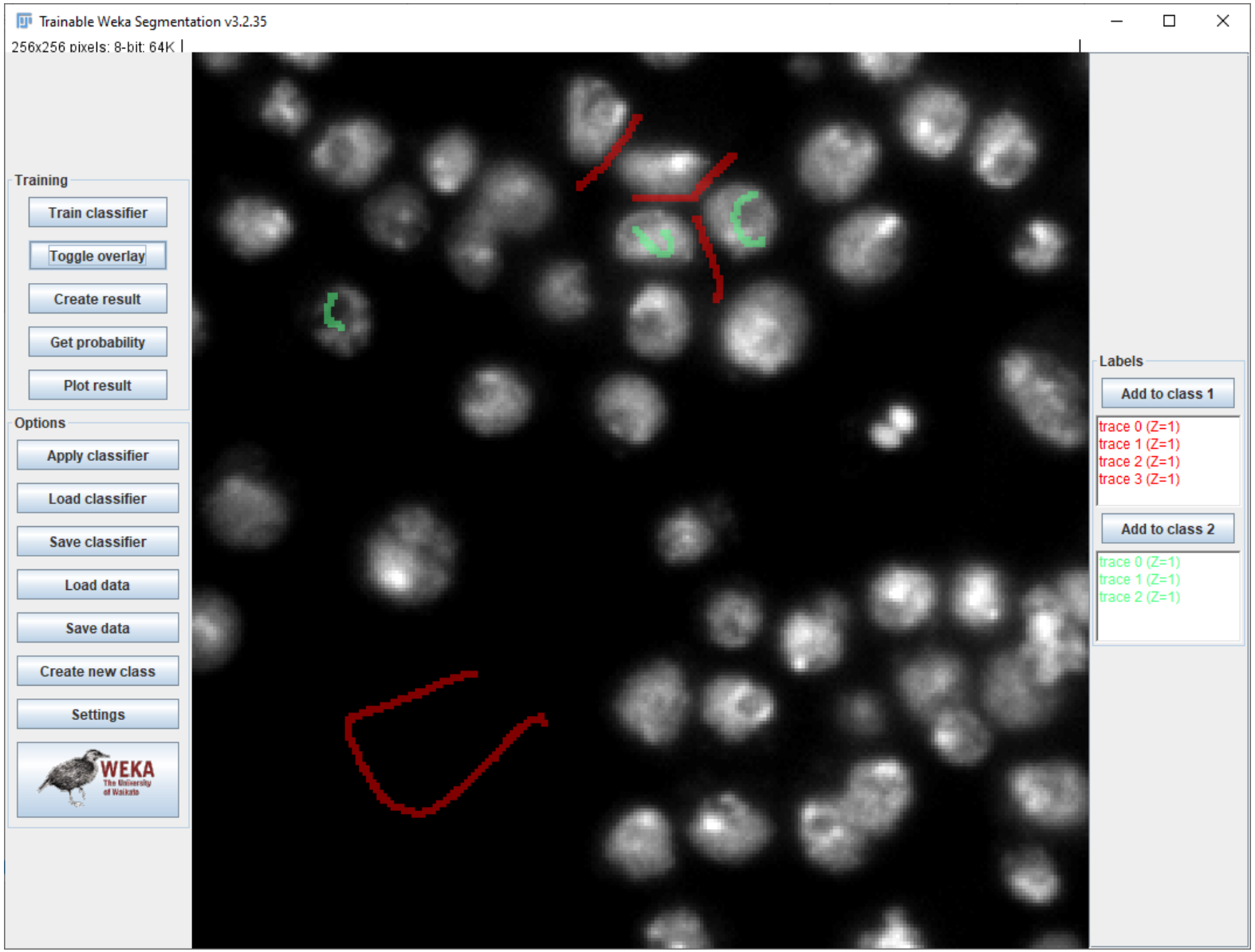


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Random Forest Pixel Classifiers

- In Practice

- Train an improved classifier

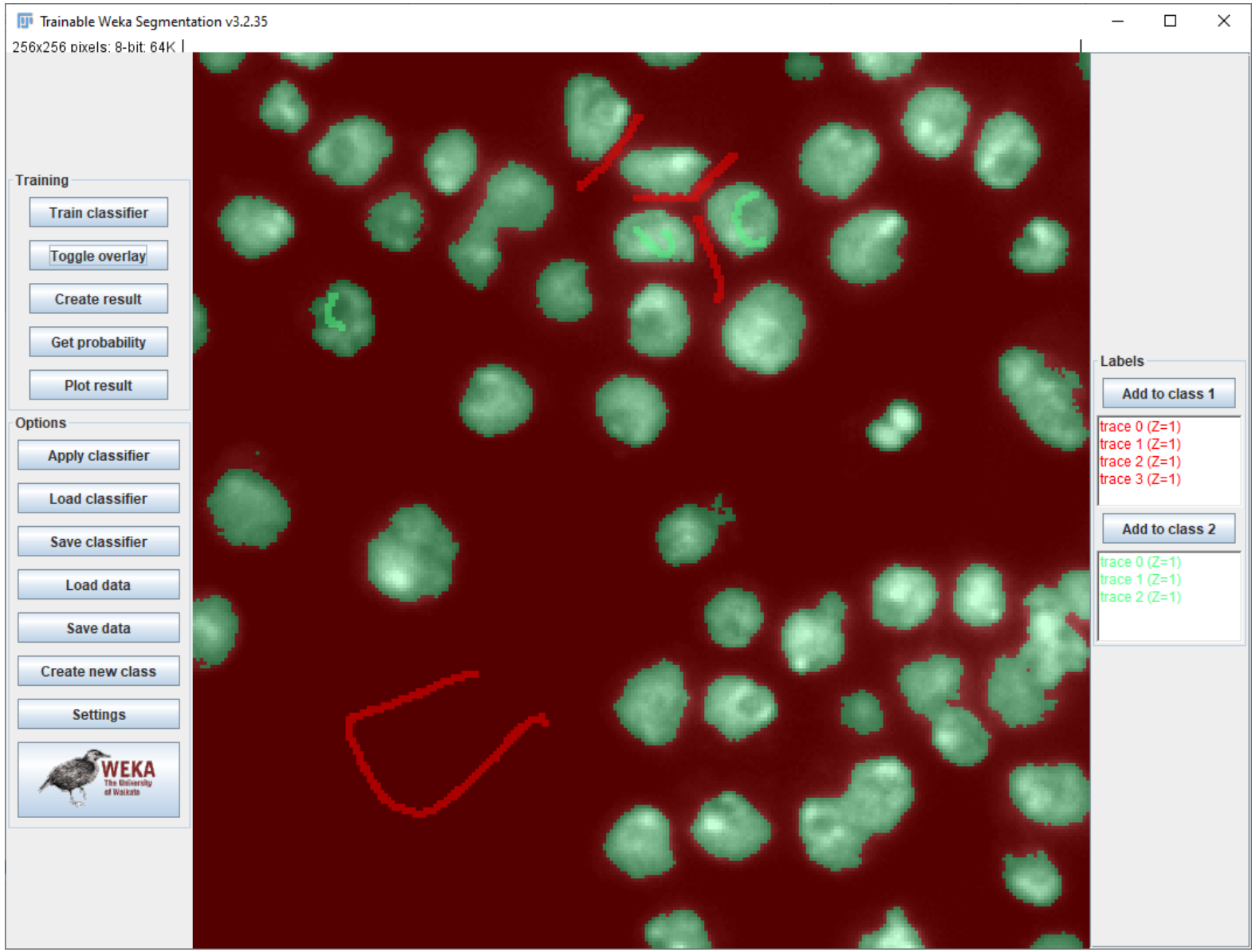
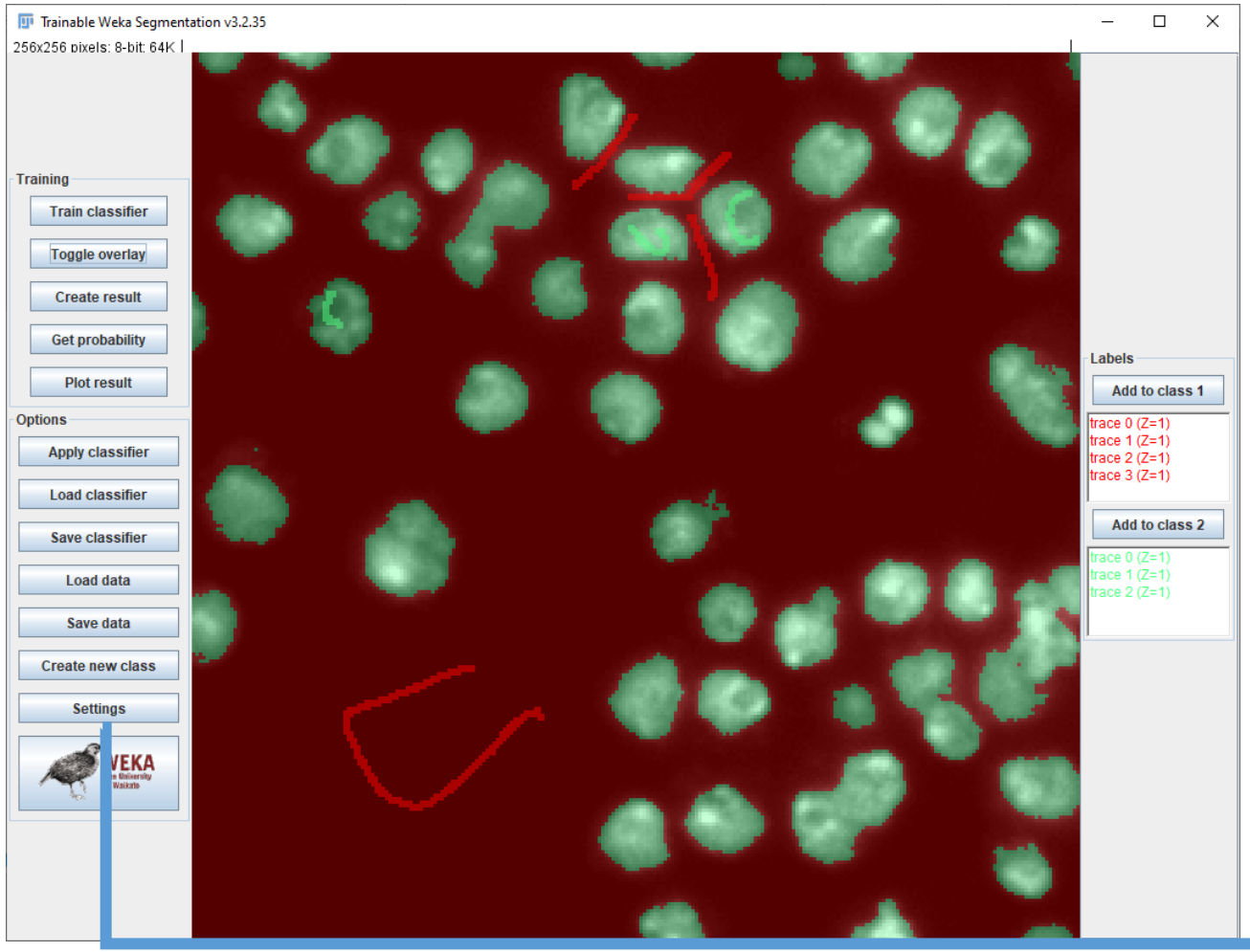
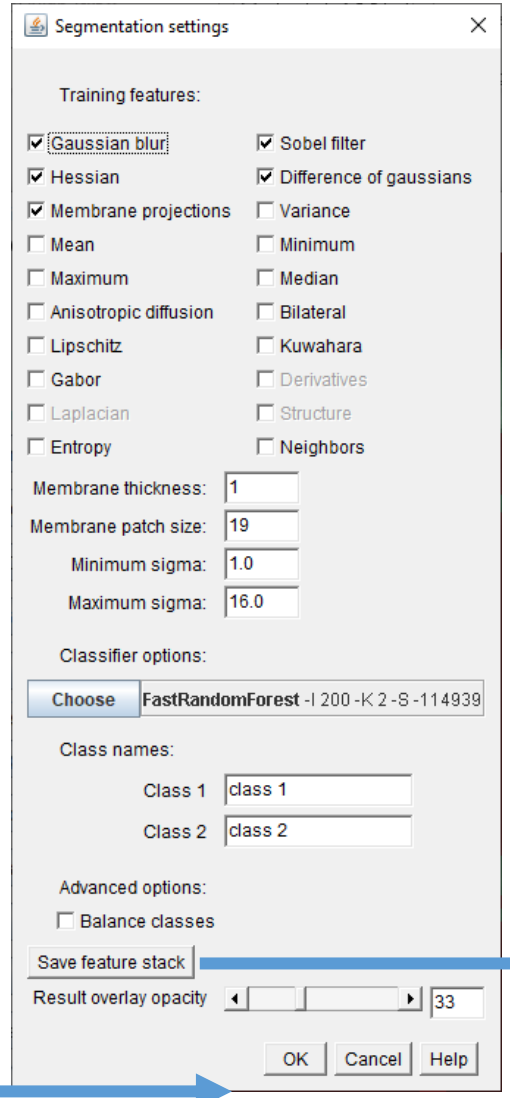


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Random Forest Pixel Classifiers

- Inspect the feature stack





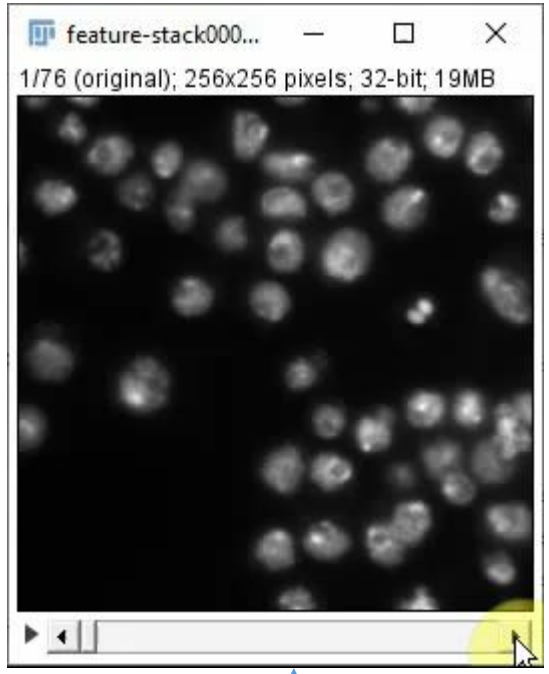
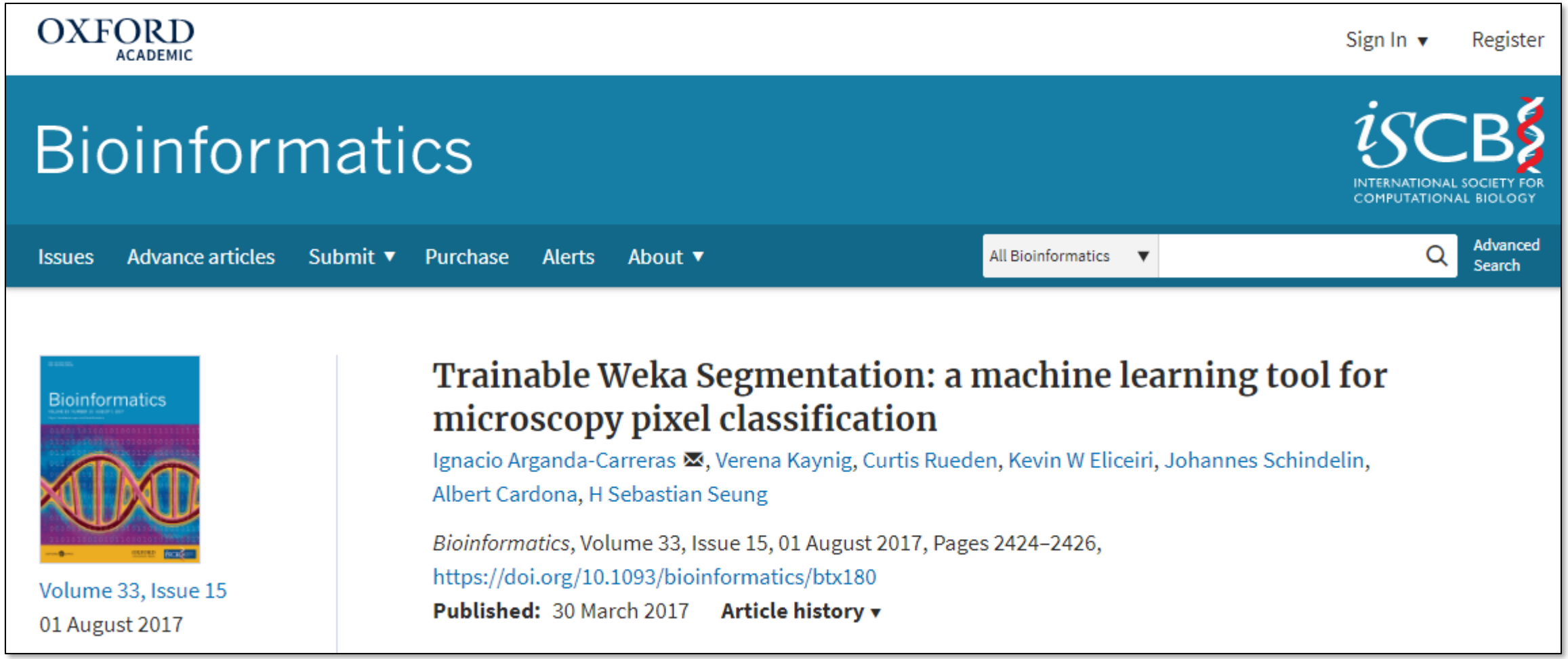


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Citation

- If you use it, please cite it:
  - https://academic.oup.com/bioinformatics/article/33/15/2424/3092362



OXFORD ACADEMIC

Sign In ▾ Register

Bioinformatics

iSCB INTERNATIONAL SOCIETY FOR COMPUTATIONAL BIOLOGY

Issues Advance articles Submit ▾ Purchase Alerts About ▾ All Bioinformatics ▾ Advanced Search

Volume 33, Issue 15
01 August 2017

Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification

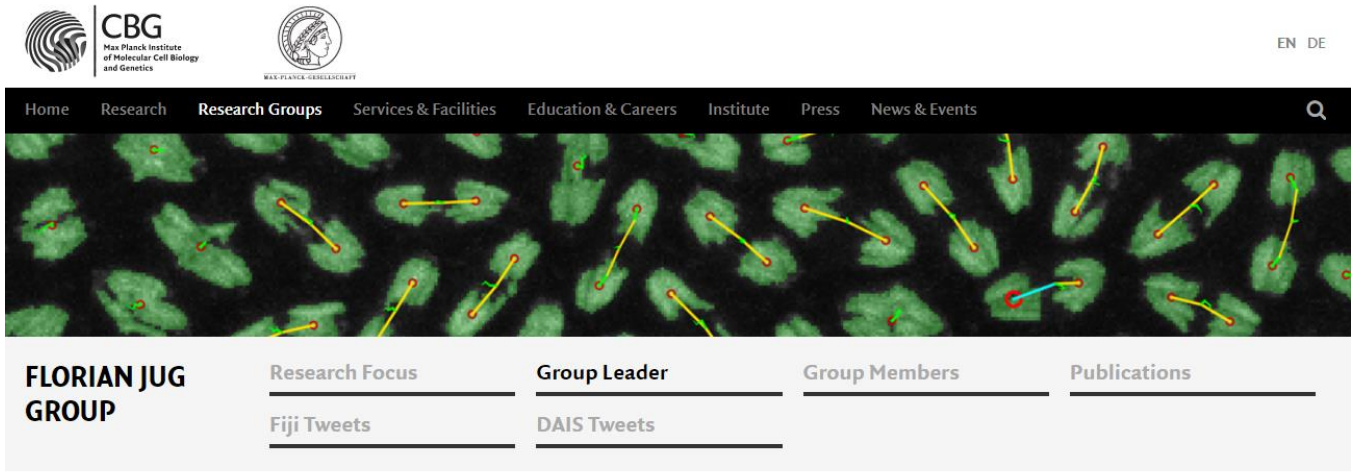
Ignacio Arganda-Carreras ✉, Verena Kaynig, Curtis Rueden, Kevin W Eliceiri, Johannes Schindelin, Albert Cardona, H Sebastian Seung

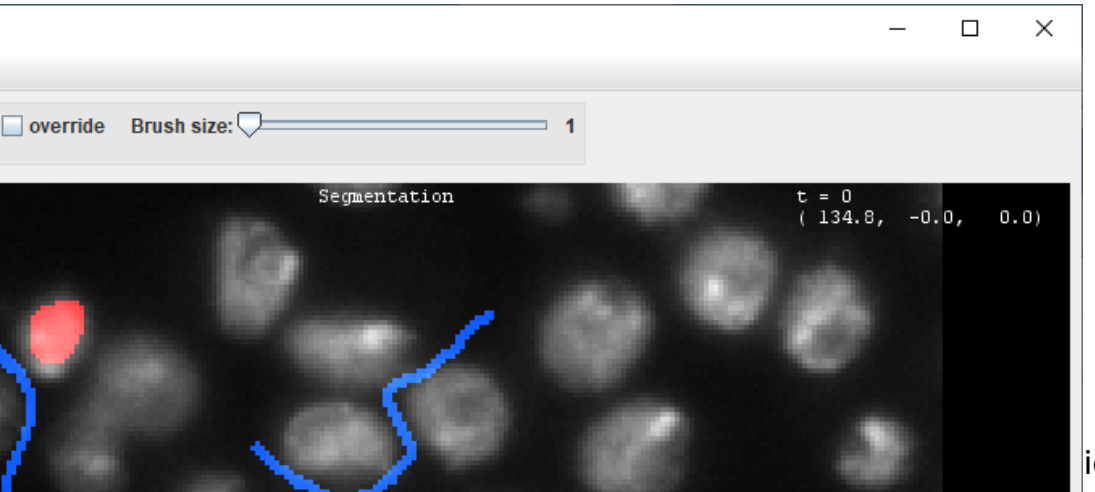
Bioinformatics, Volume 33, Issue 15, 01 August 2017, Pages 2424–2426,
https://doi.org/10.1093/bioinformatics/btx180

Published: 30 March 2017 **Article history** ▾

LabKit

- “Labeling kit”: A Fiji plugin for annotating objects in big image data (> larger than RAM)
- Weka based pixel classification





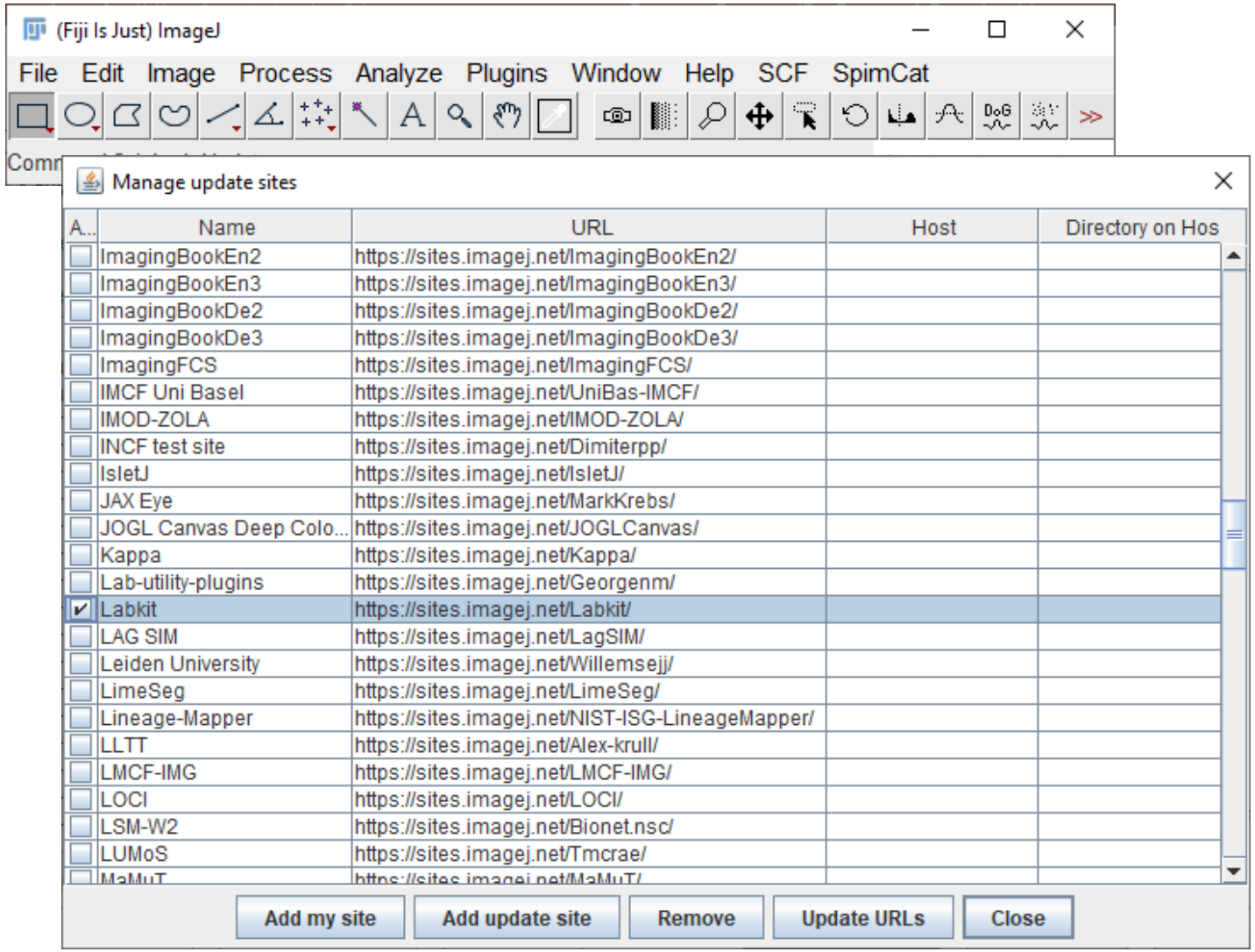
FLORIAN JUG

Matthias Arzt, RSE

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

LabKit

- Help > Update…. Manage update sites….



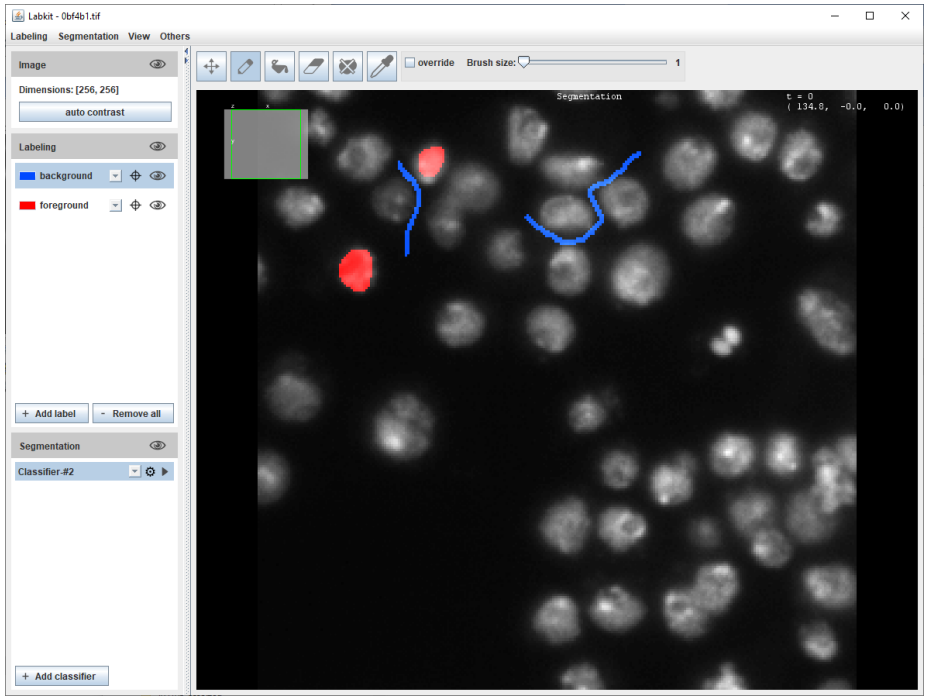
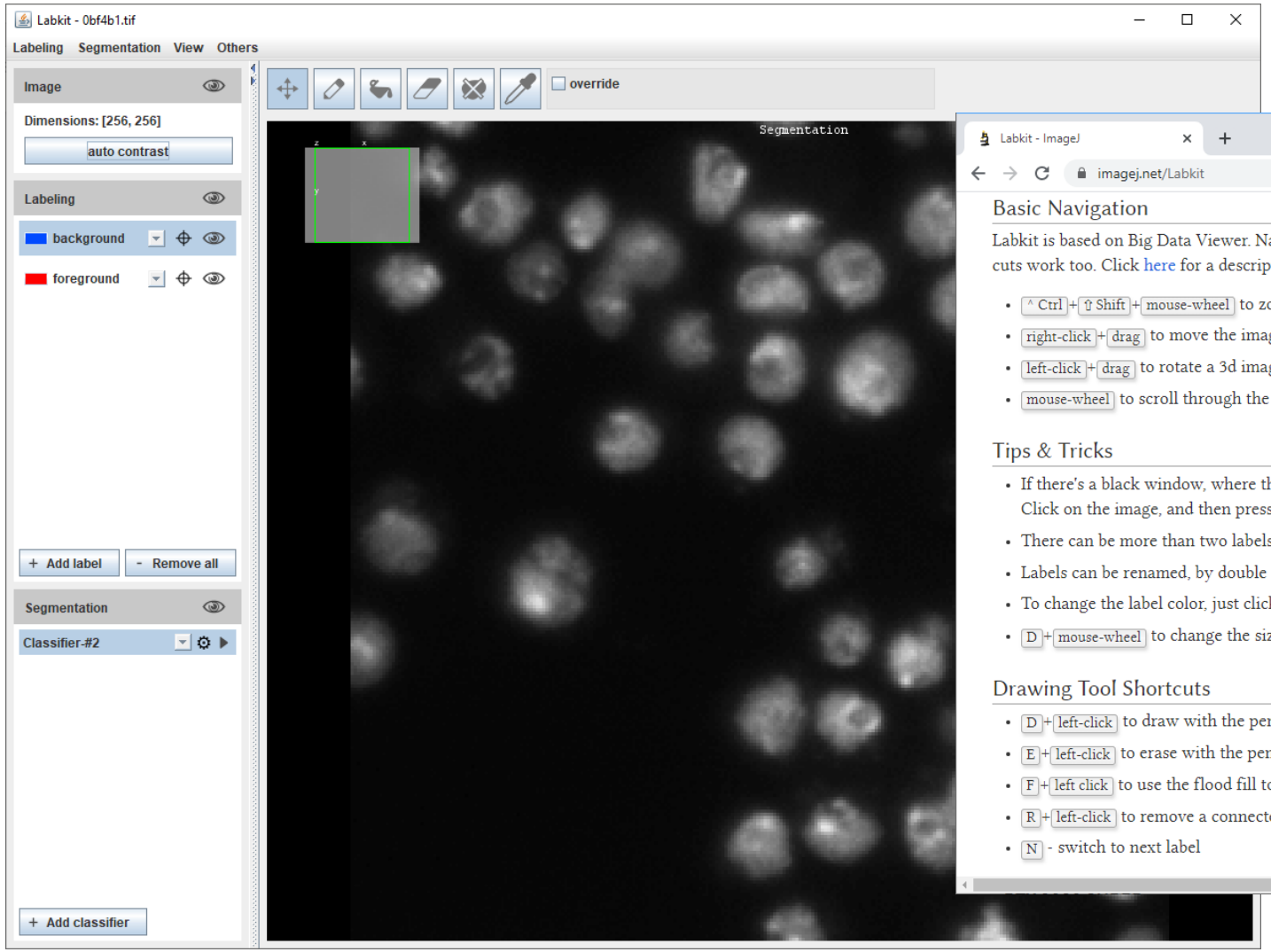
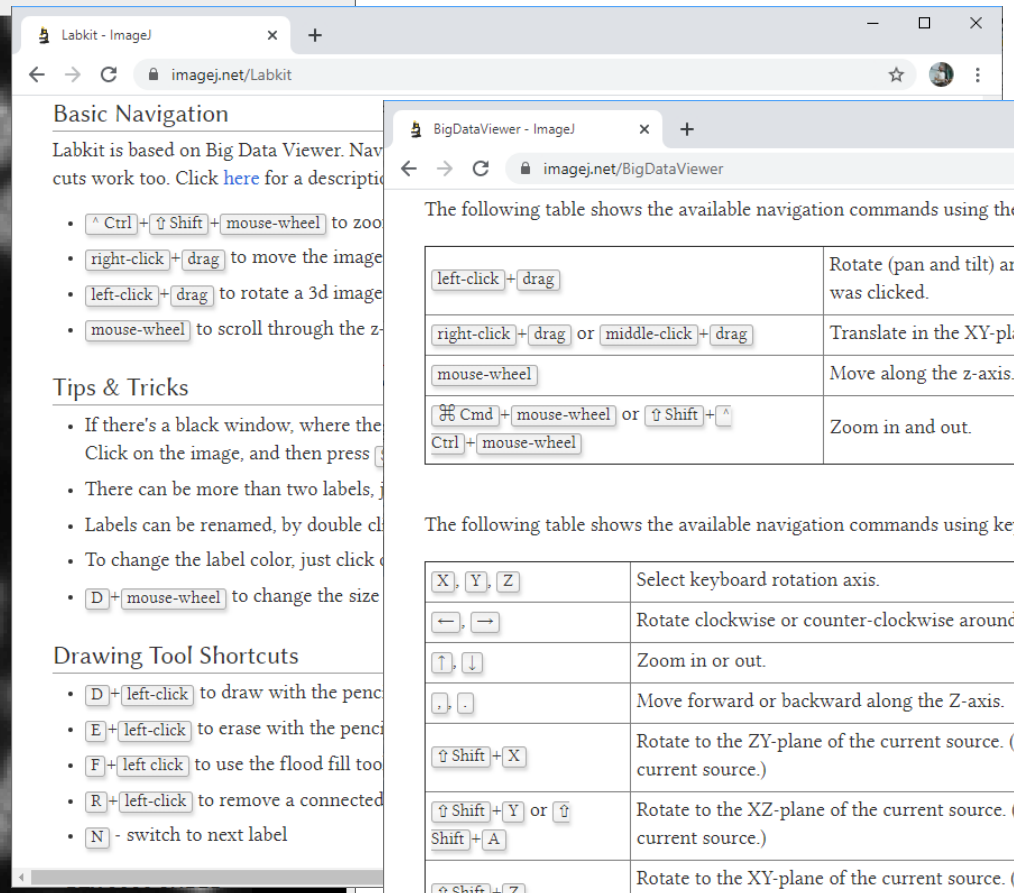


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

LabKit

- Plugins > Segmentation > LabKit





- Help:
  - https://imagej.net/Labkit
  - https://imagej.net/BigDataViewer

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

LabKit

- Annotations: Foreground vs. background vs. …

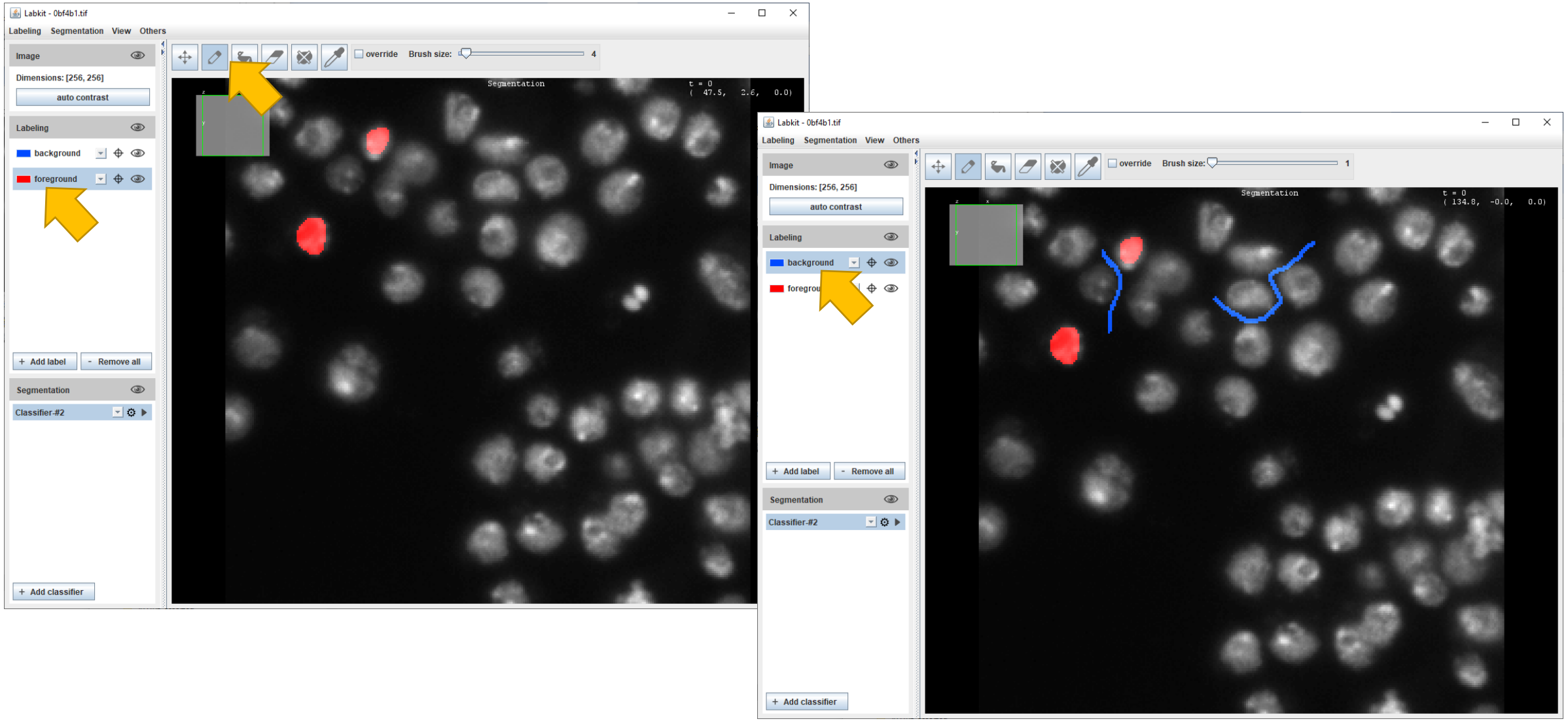


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

LabKit

- Feature selection + Training

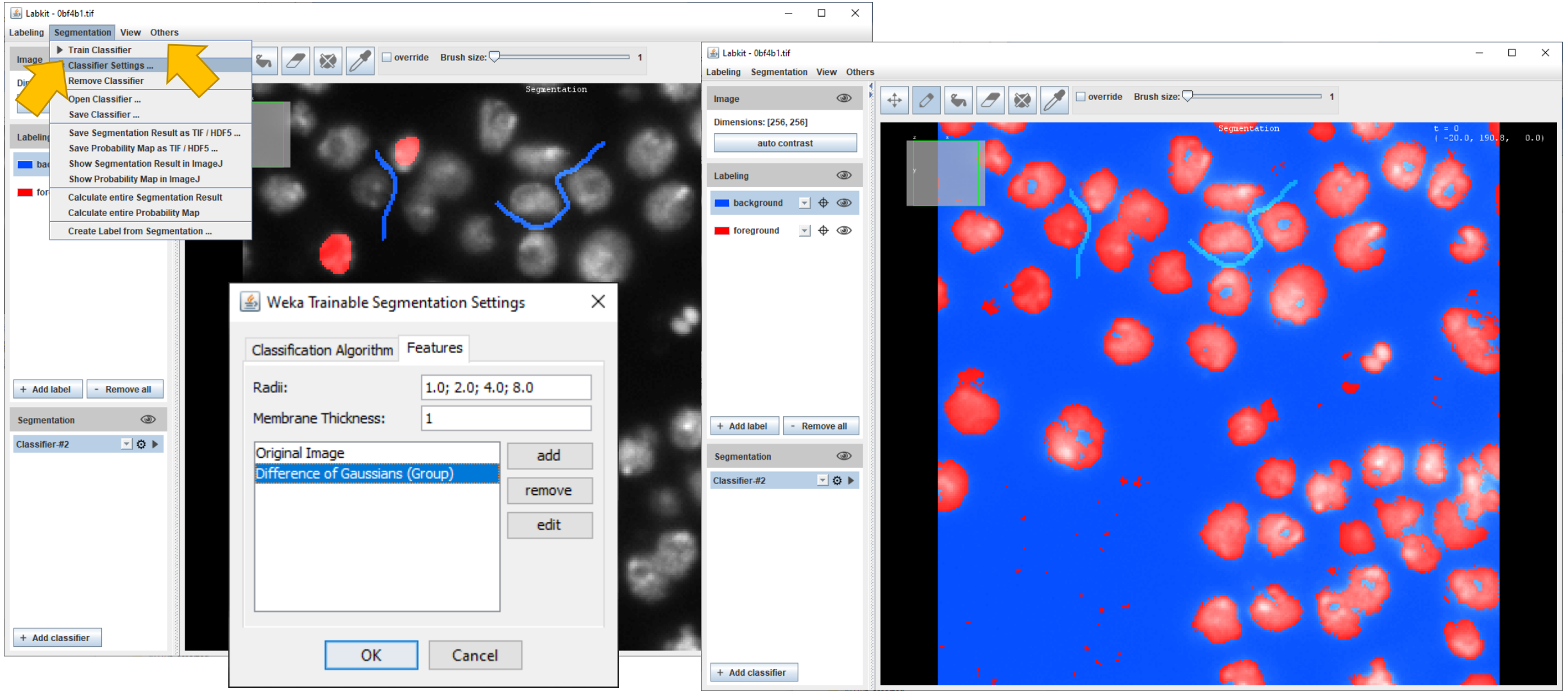
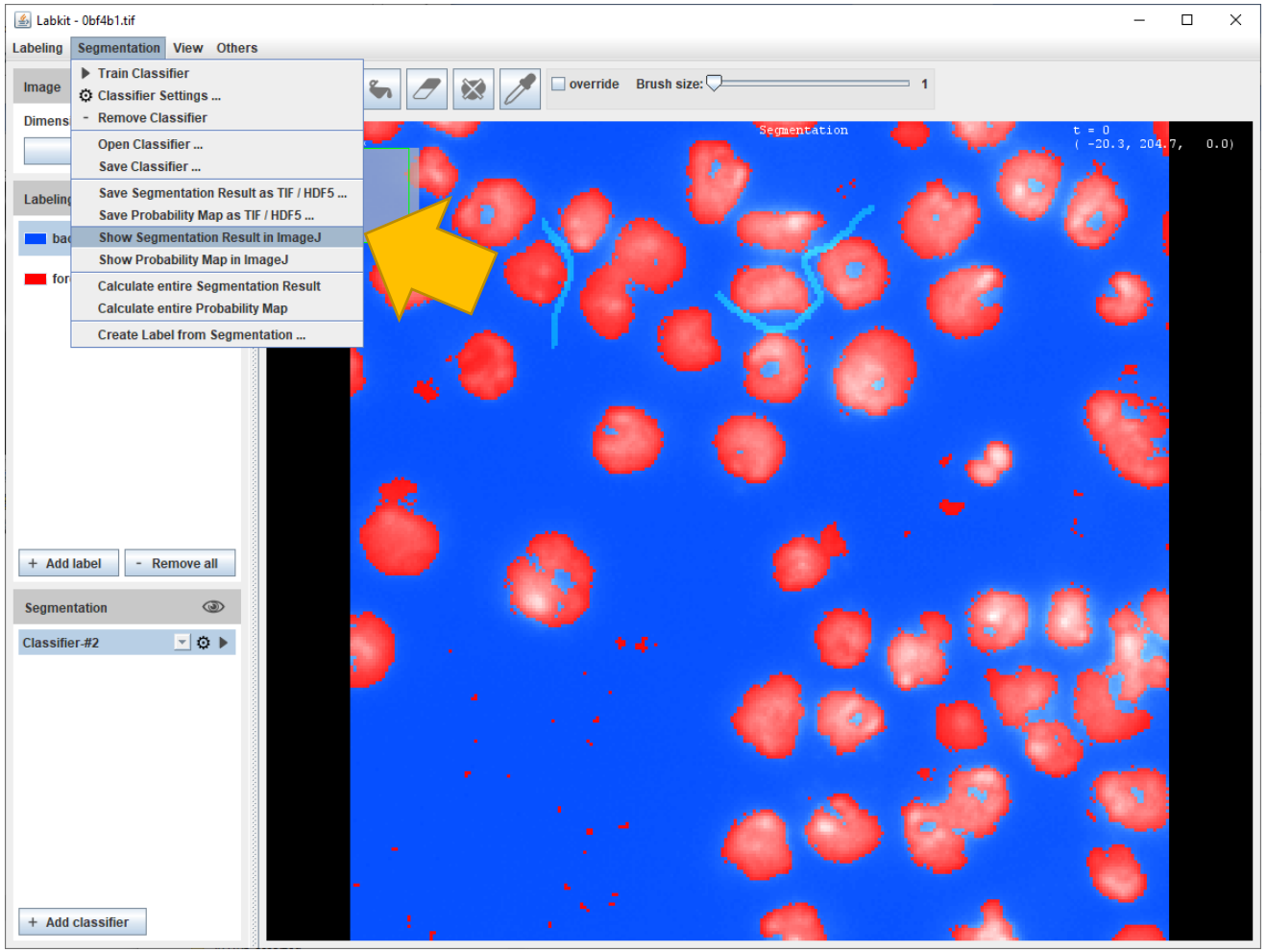
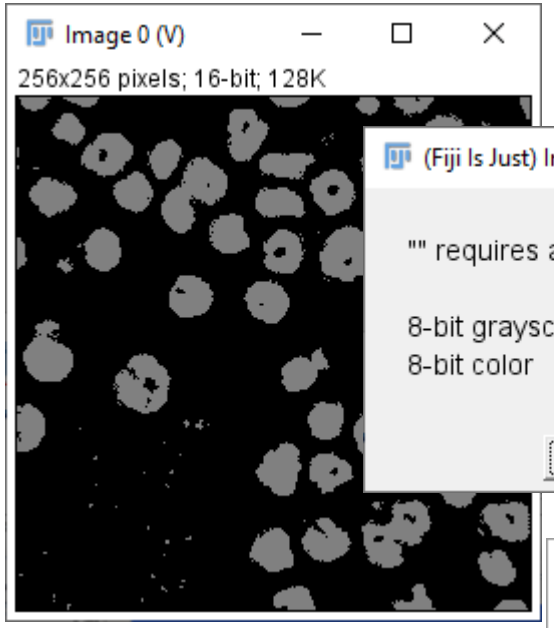


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

LabKit

- Exporting and post-processing: Exported binary images need to be multiplied by 255 and converted to 8-bit





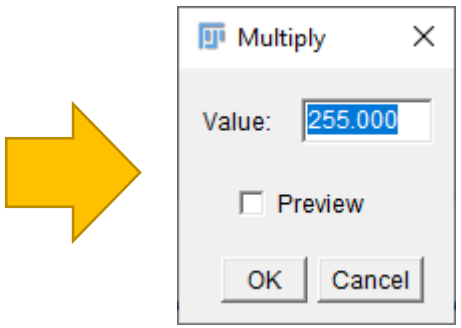
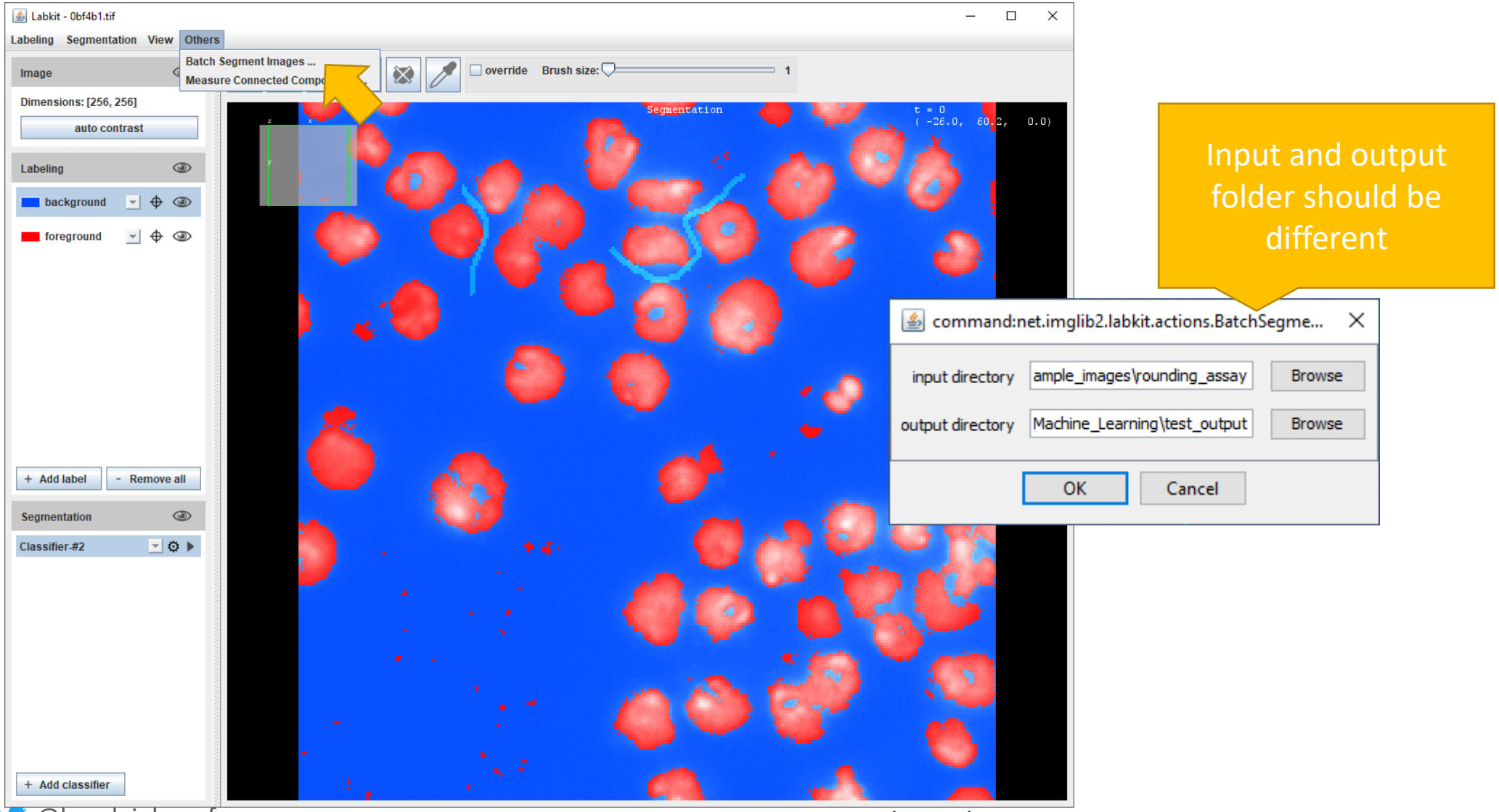


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

LabKit

- Generate predictions of a folder of images

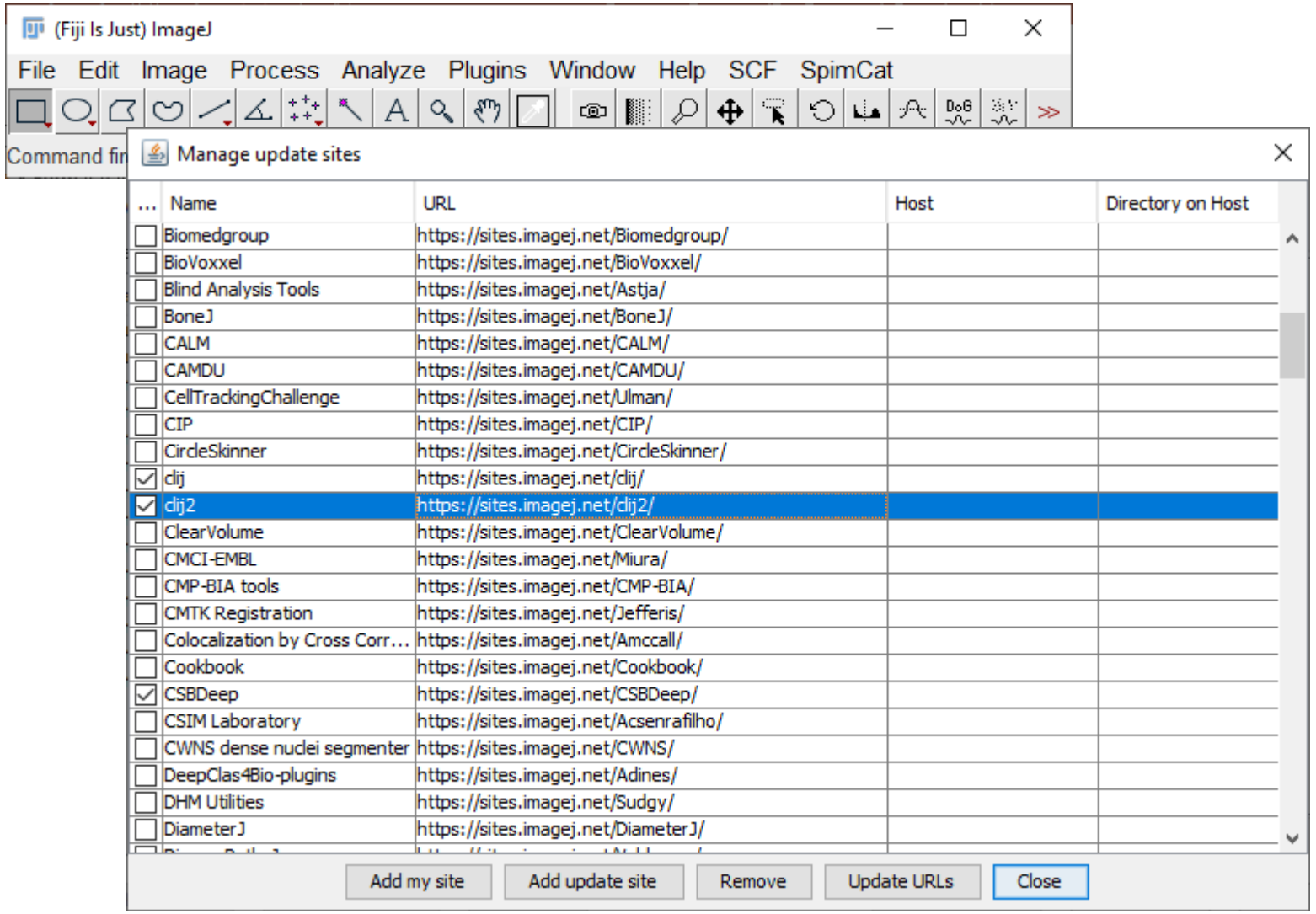


Input and output folder should be different

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

CLIJxWeka Object classification

- An academic machine learning tool for 2D Object classification in Fiji



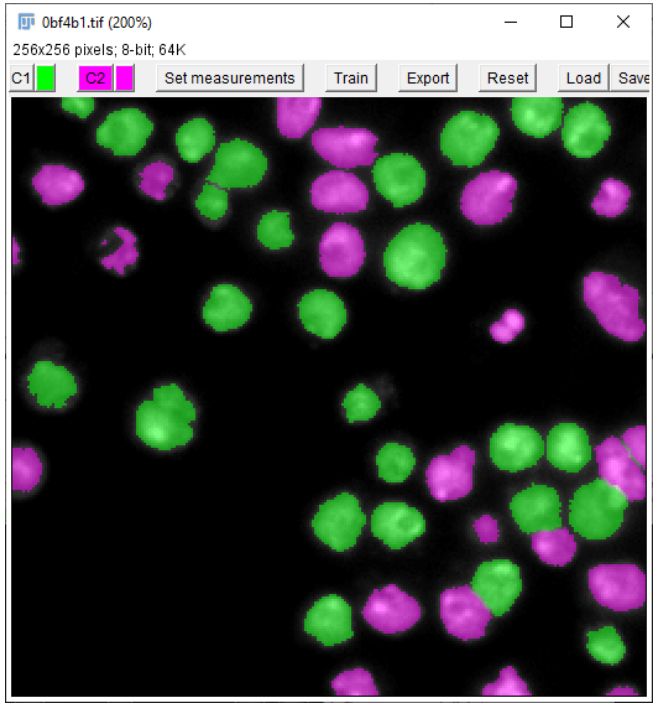
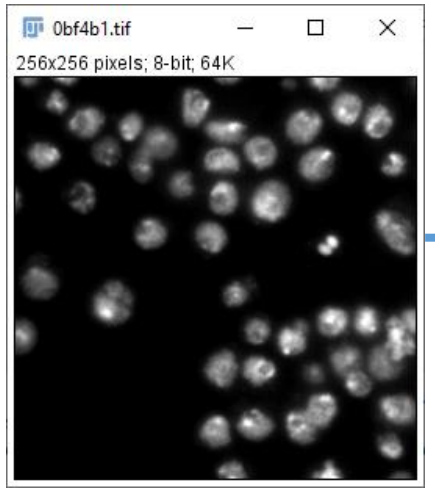
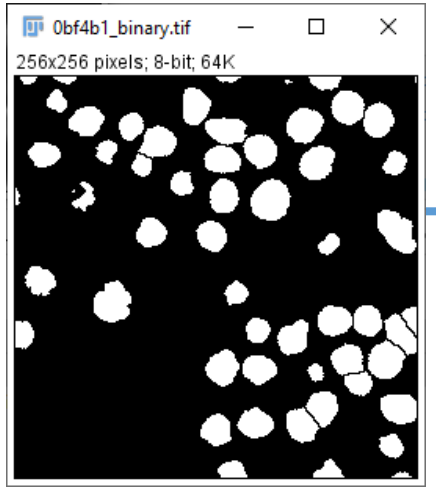


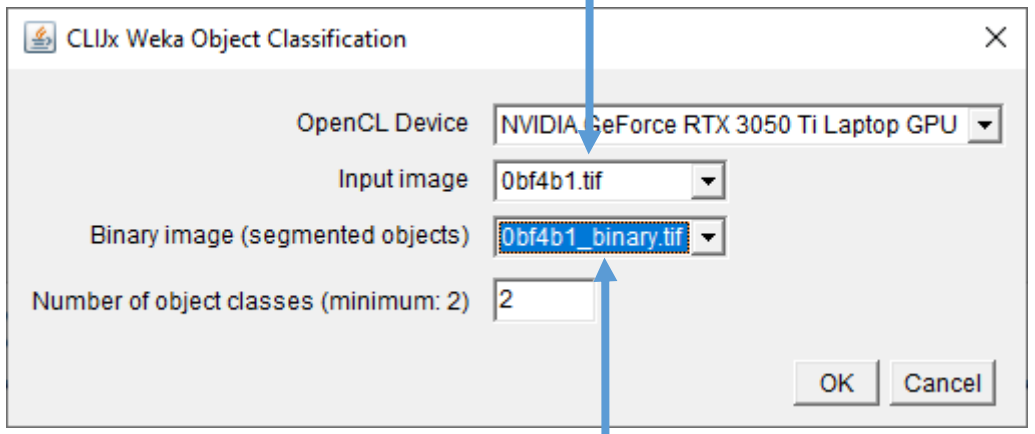
Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

CLIJxWeka Object classification

- Input: Original image + binary image
- Plugins > Segmentation > CLIJx Weka Object Classification on GPU (CLIJx, experimental)







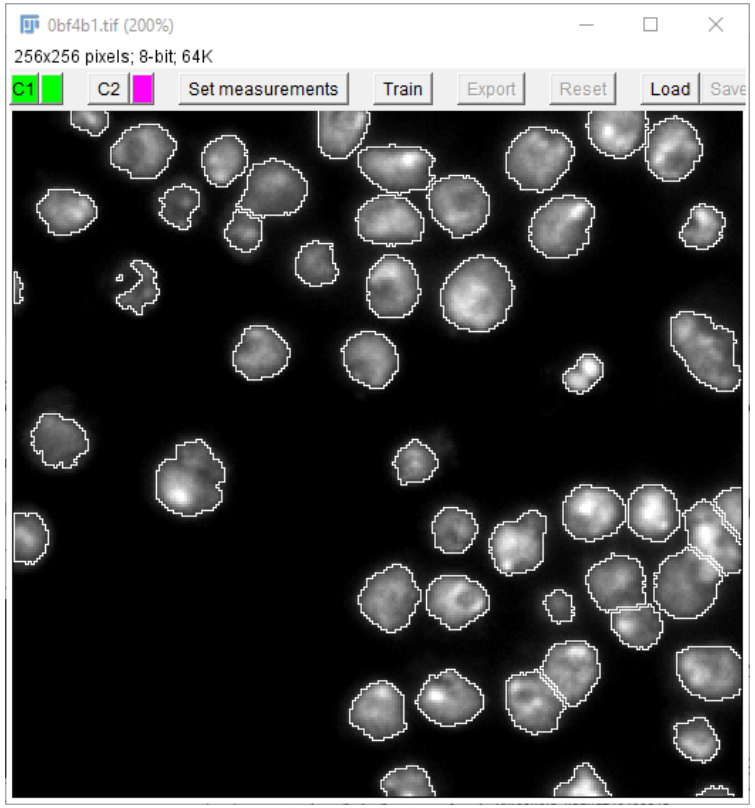
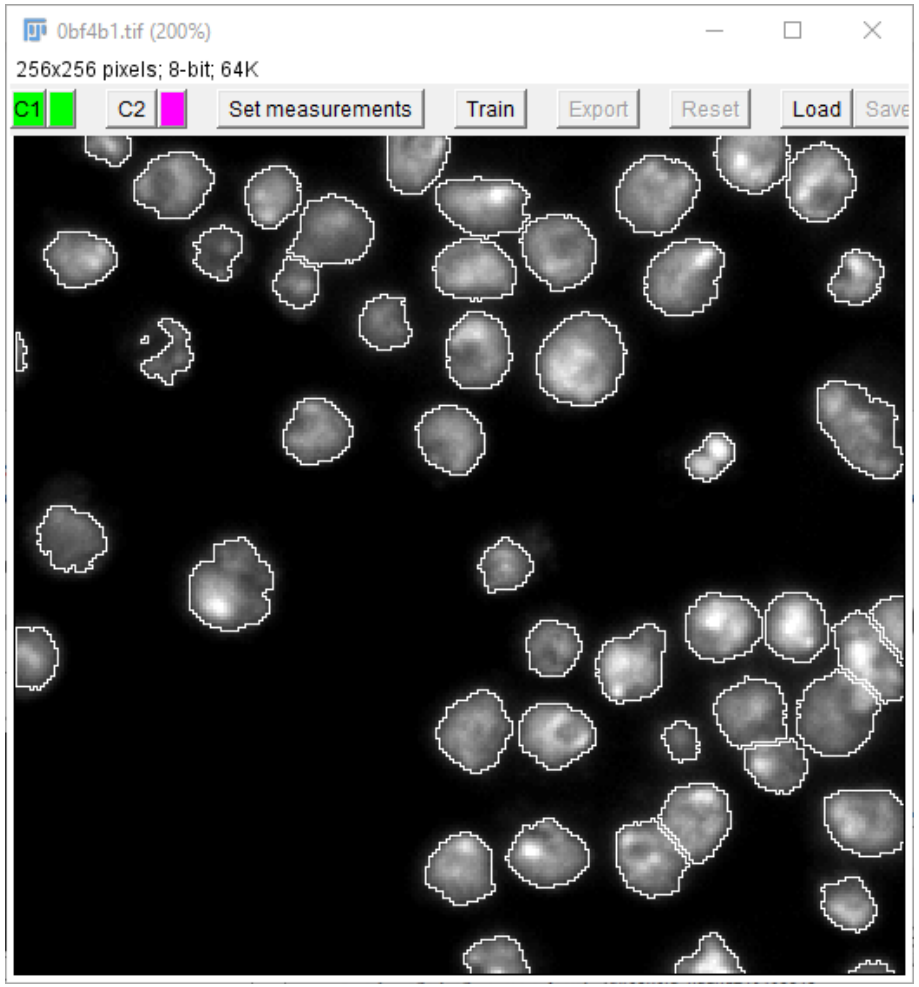
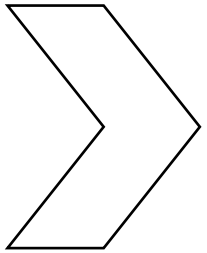


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

CLIJxWeka Object classification

- Add annotation: Click
- Remove annotation: Click again





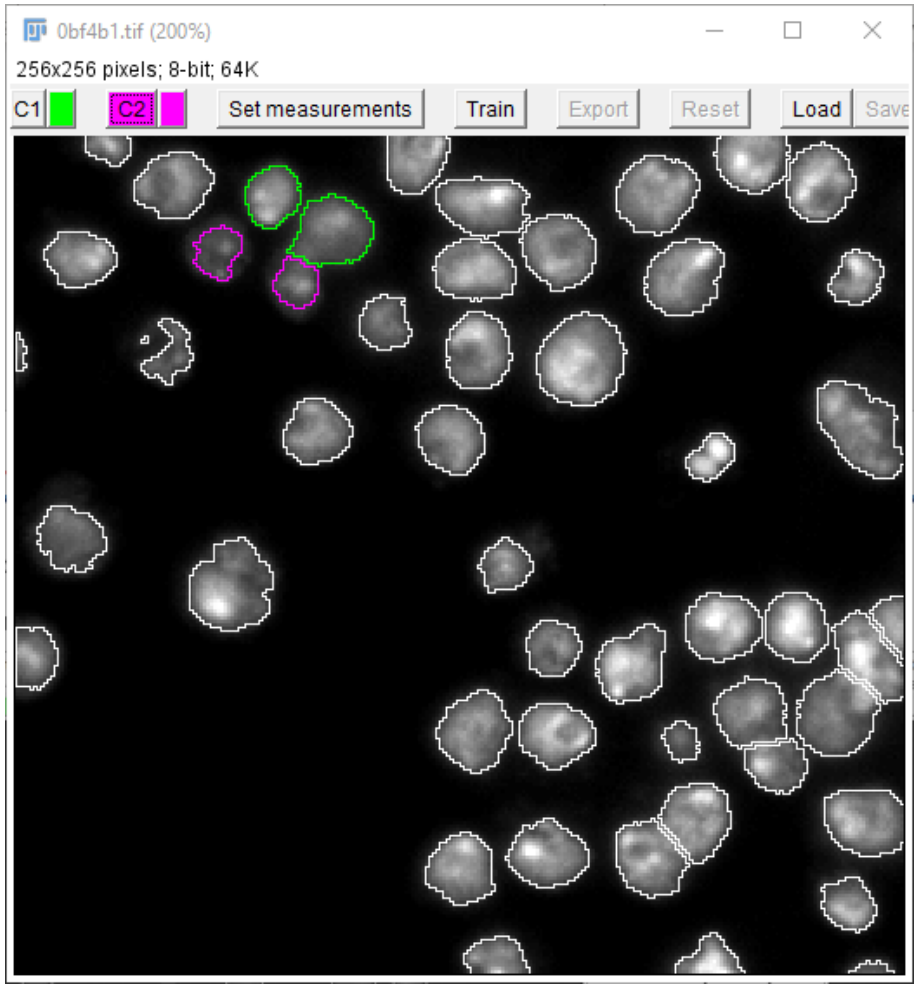


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

CLIJxWeka Object classification

- Feature selection + training

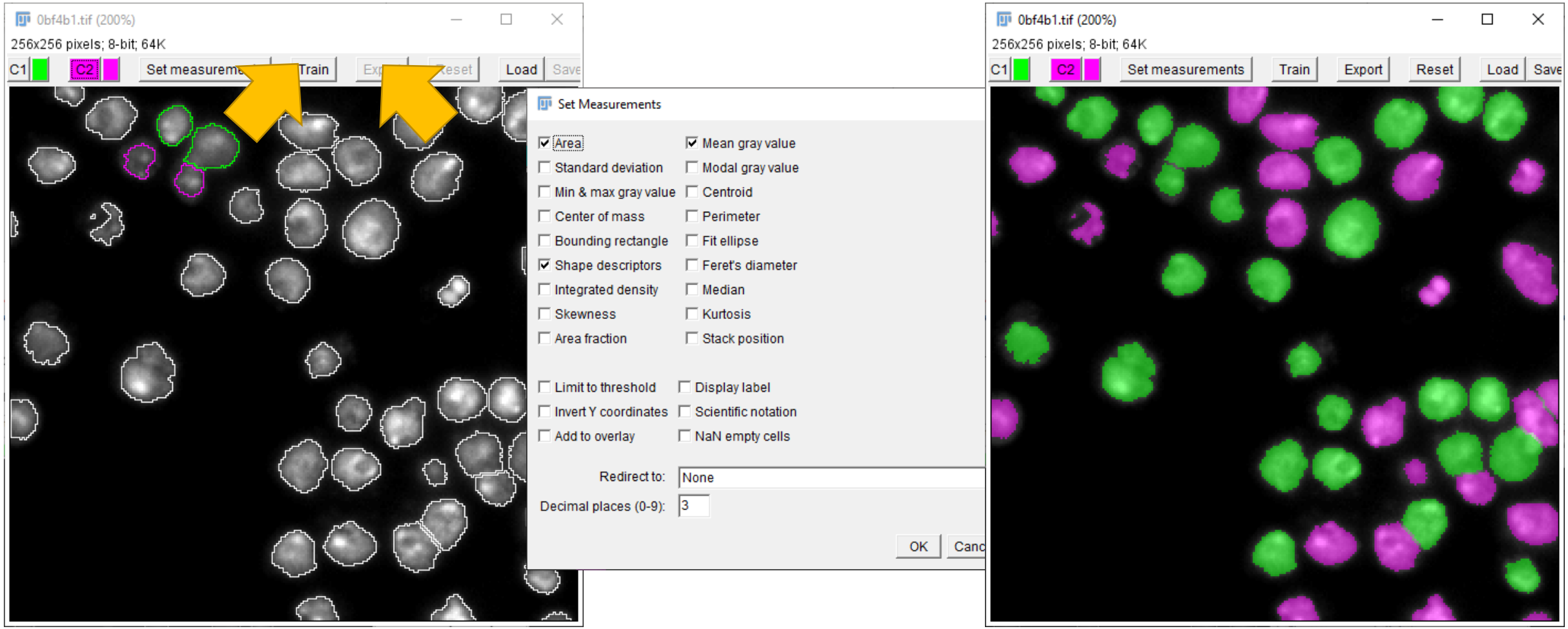


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Image Segmentation in Python using scikit-learn

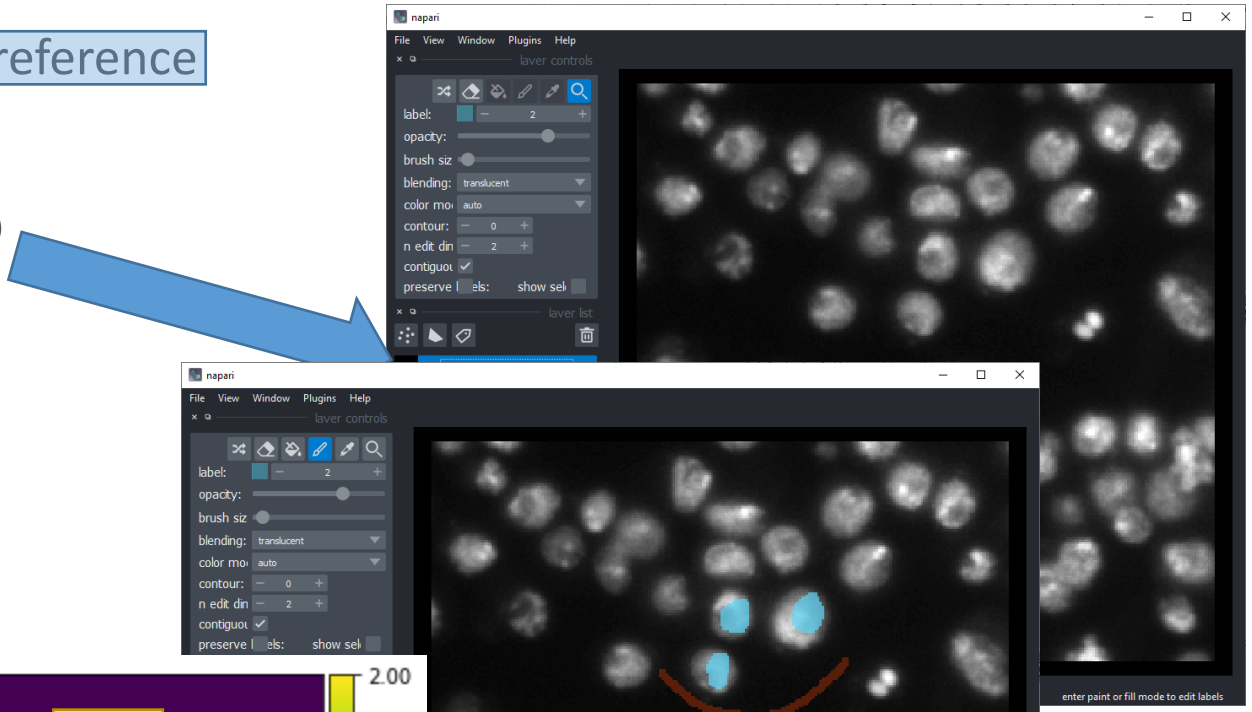
Robert Haase

May 2021

Interactive pixel classification

- Prepare an empty layer for annotations and keep a reference

```
labels = viewer.add_labels(
    np.zeros(image.shape).astype(int))
```



- Read annotations

```
manual_annotations = labels.data
```

```
from skimage.io import imshow
imshow(manual_annotations,
               vmin=0, vmax=2)
```

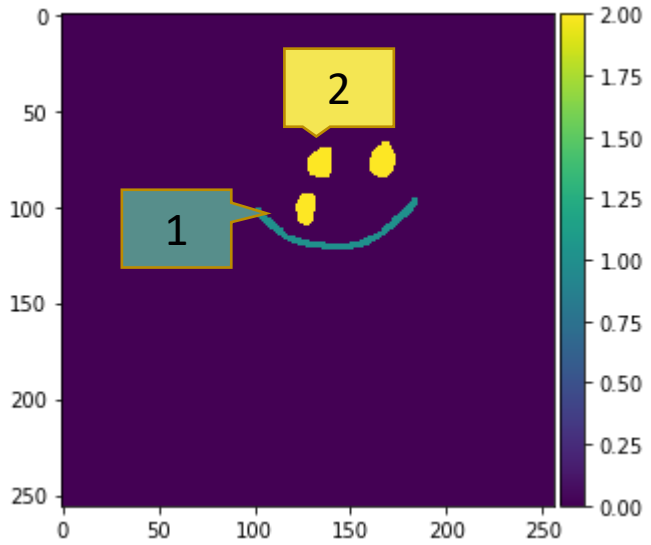


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Interactive pixel classification

- Pixel classification using scikit-learn
  - Expects one-dimensional arrays for
    - every feature individually
    - ground truth

```
# train classifier
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(max_depth=2, random_state=0)
classifier.fit(X, y)
```

Image data

Ground truth / annotation

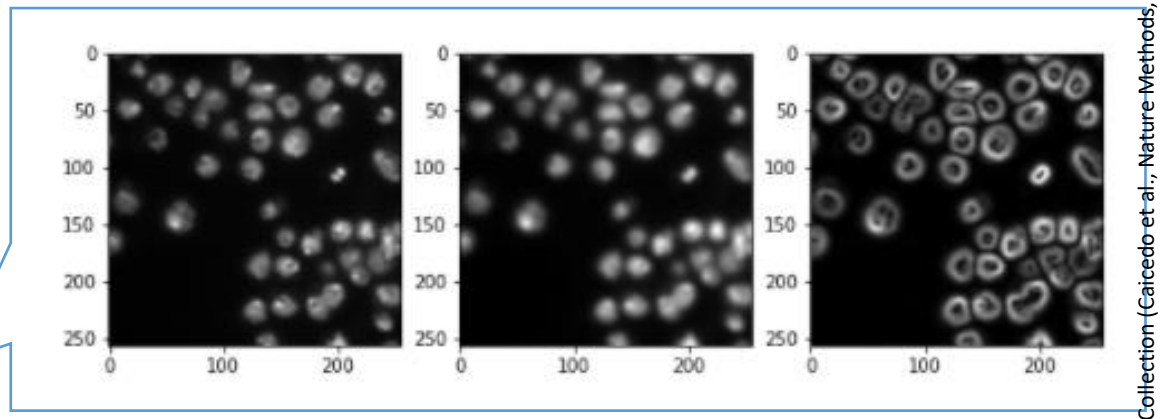
```
y_ = classifier.predict(X)
```

Image data

prediction

Interactive pixel classification

- Pixel classification using scikit-learn
  - Expects one-dimensional arrays for
    - every feature individually
    - ground truth




```
# for training, we need to generate features
feature_stack = generate_feature_stack(image)
X, y = format_data(feature_stack, manual_annotations)

# train classifier
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(max_depth=2, random_state=0)
classifier.fit(X, y)
```

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Interactive pixel classification

- Pixel classification using scikit-learn

```
# process the whole image and show result
result_1d = classifier.predict(feature_stack.T)
result_2d = result_1d.reshape(image.shape)

viewer.add_labels(result_2d)
```

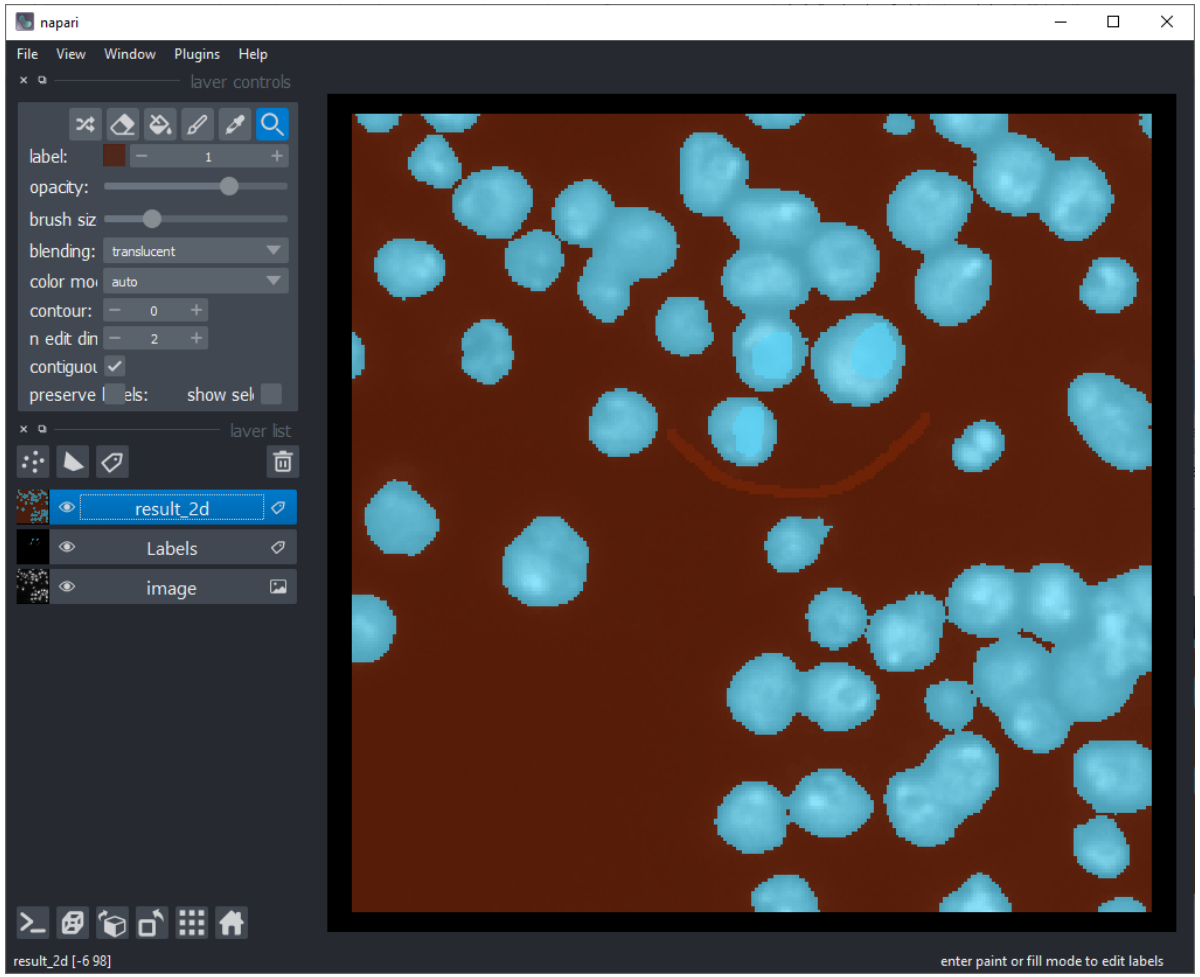


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Interactive pixel classification

- Jupyter notebooks and napari side-by-side

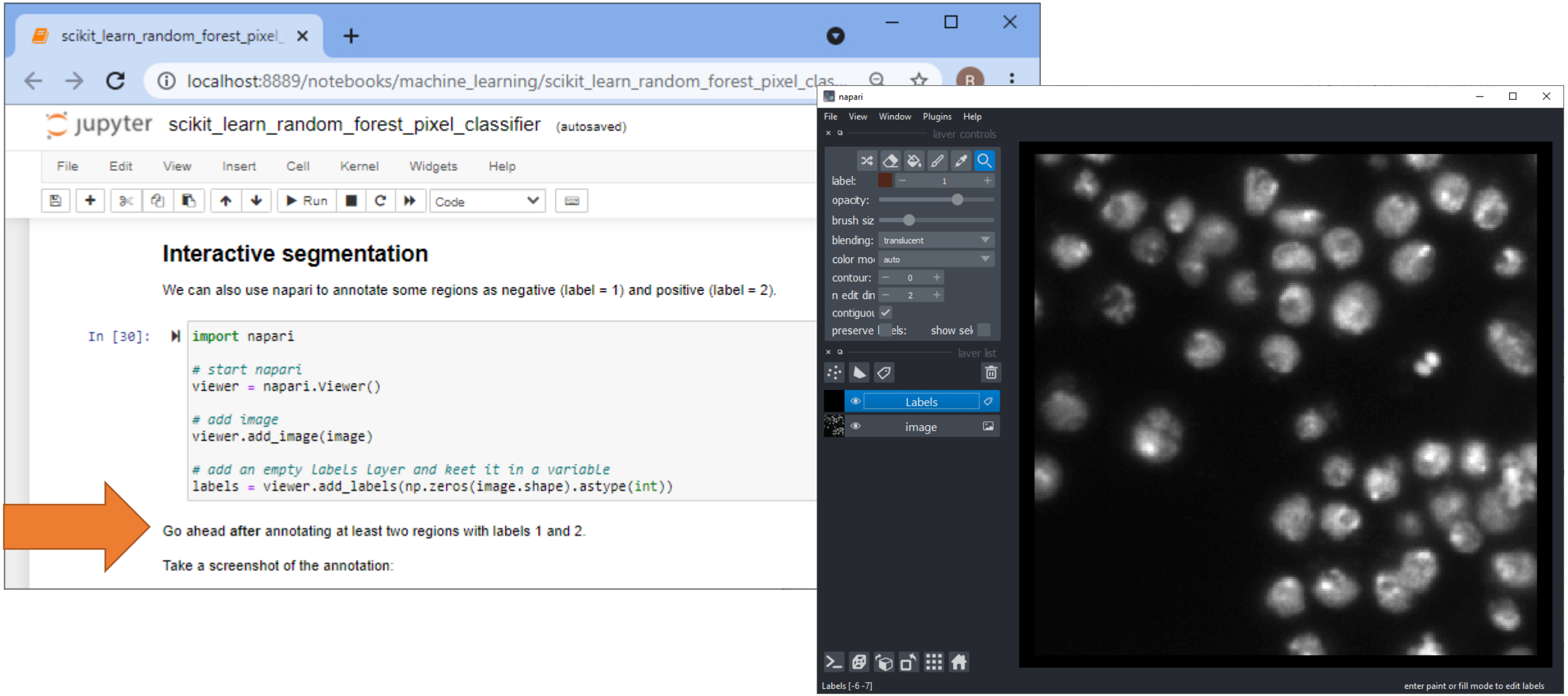


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Interactive pixel classification

- Jupyter notebooks and napari side-by-side

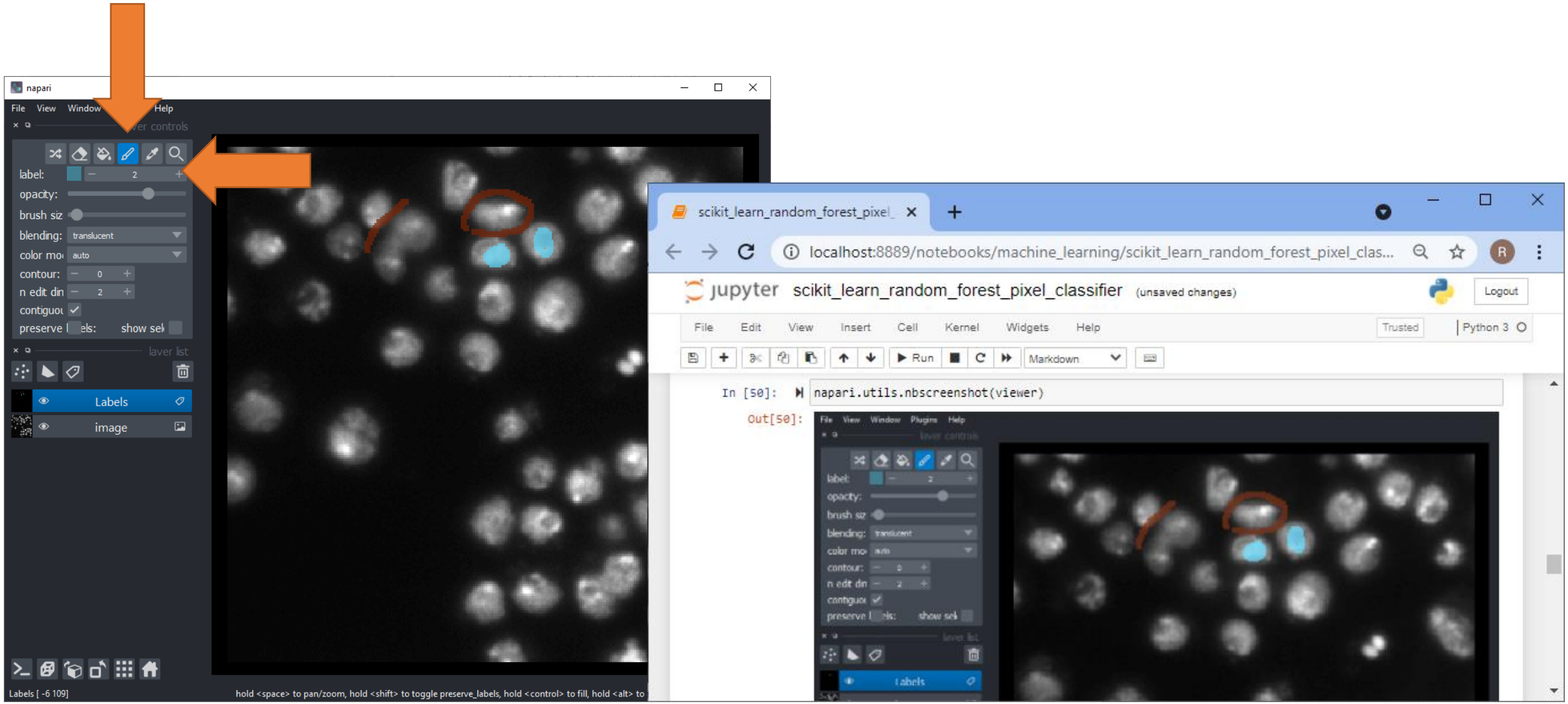


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Interactive pixel classification

- Jupyter notebooks and napari side-by-side

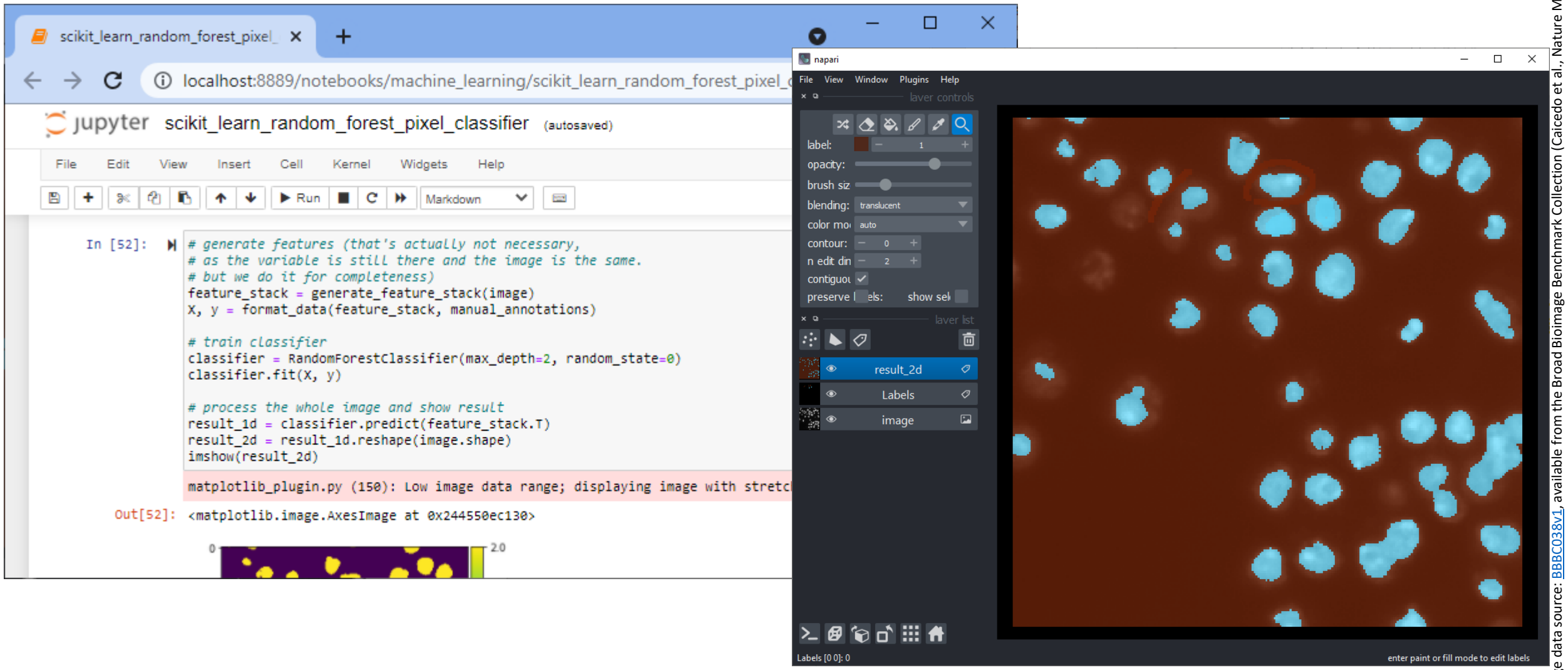


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

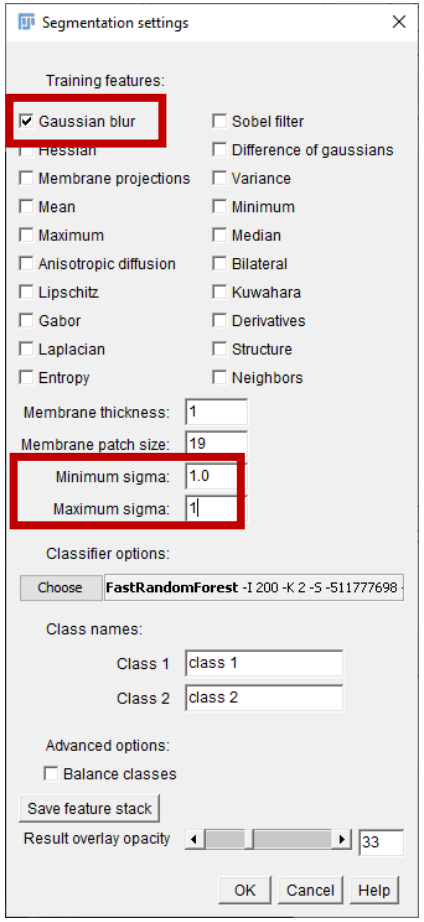
PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Exercises

Exercises: Pixel classification using Trainable Weka / Labkit

- Use Trainable Weka Segmentation and / or Labkit
- Download an example image from the Broad BioImage Benchmark collection:
  - https://bbbc.broadinstitute.org/BBBC038
  - Use image 0bf4b144167694b6846d584cf52c458f34f28fcae75328a2a096c8214e01c0d0
- Train a random forest classifier using Weka Trainable Segmentation or LabKit in Fiji
- Can a classier with just a single Gaussian blur be trained successfully?
- If yes, what does that imply?
- Is machine learning necessary to solve this segmentation problem?



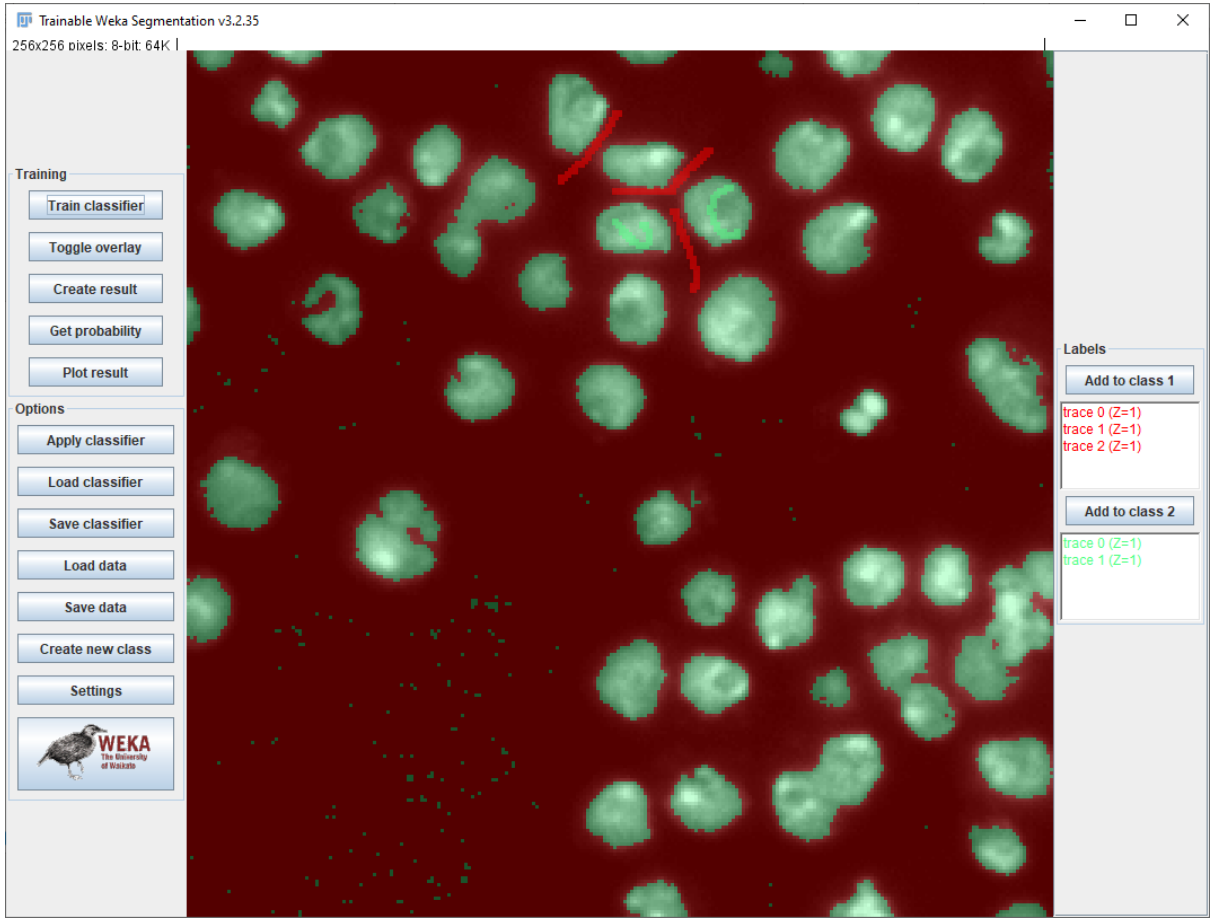
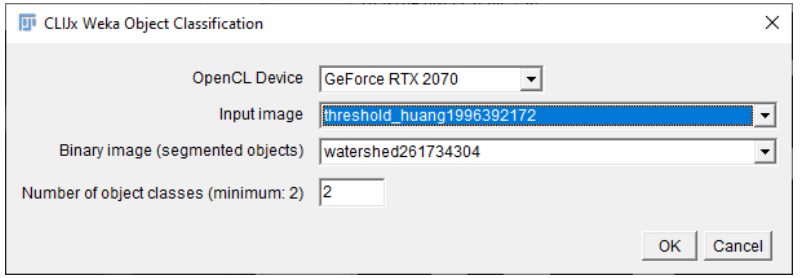
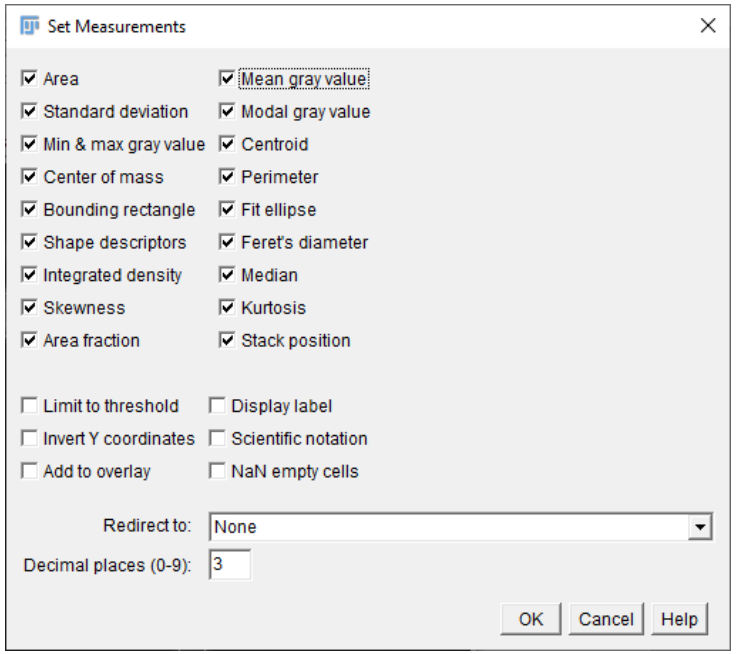


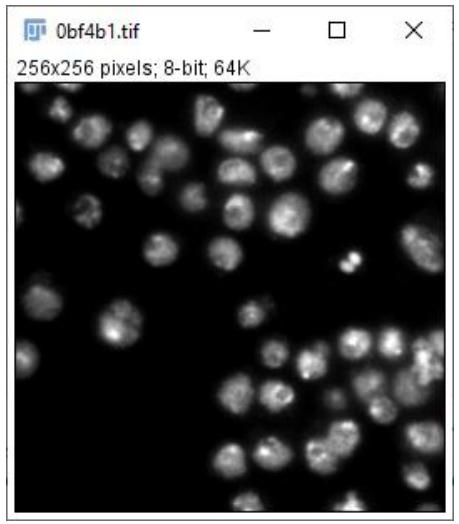
Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

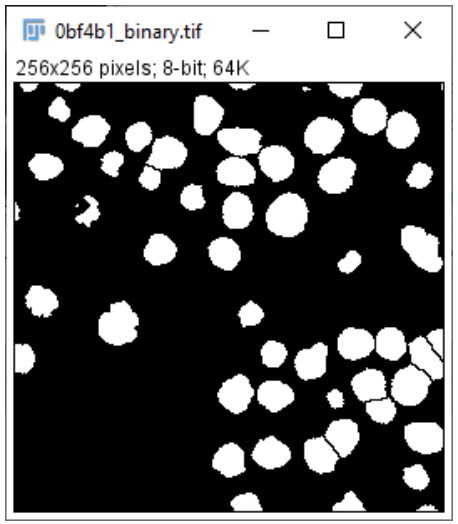
Exercise: Object classification

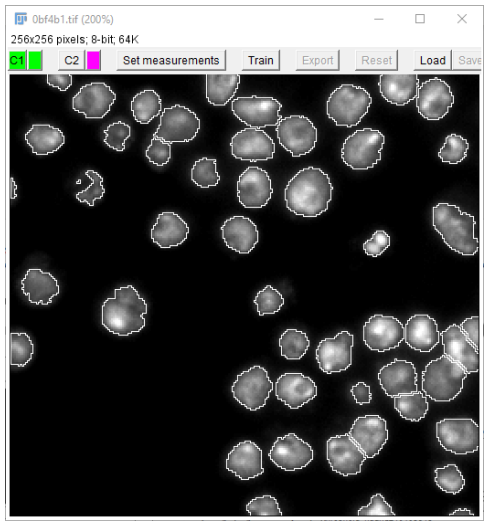
- Test CLIJx Weka Object classification to separate objects according to
  - Position
  - Shape
  - Size
- Which measurements do you have to set?
- Optional: Can you classify three classes?
  - Small, large, non-round

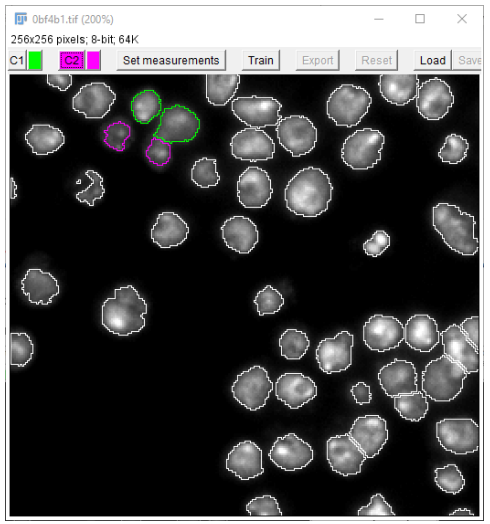












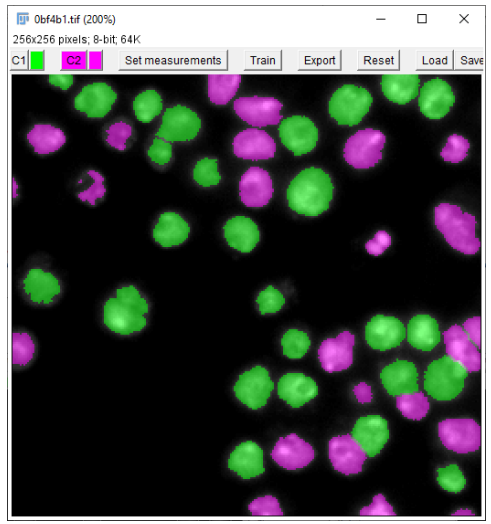
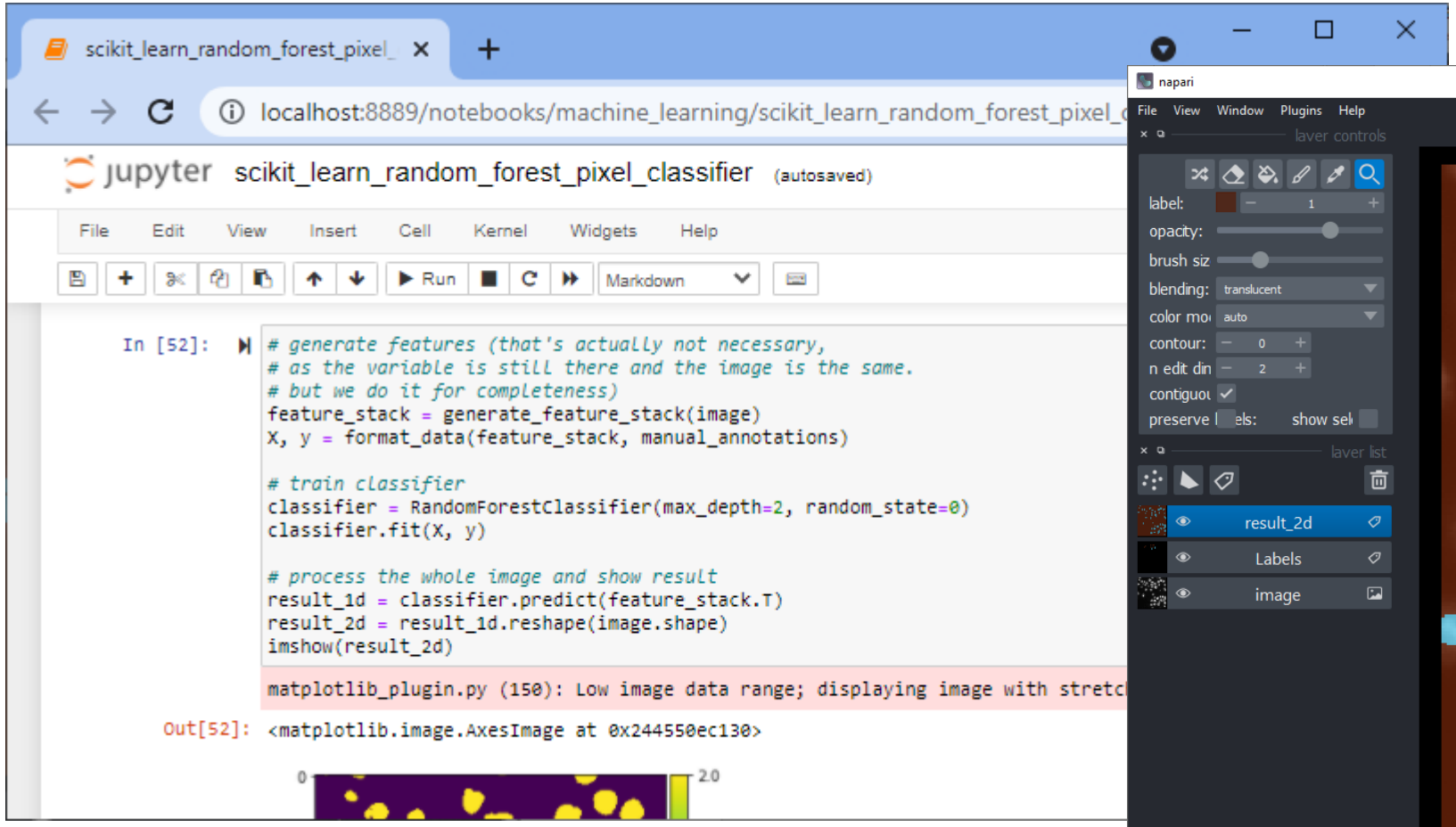
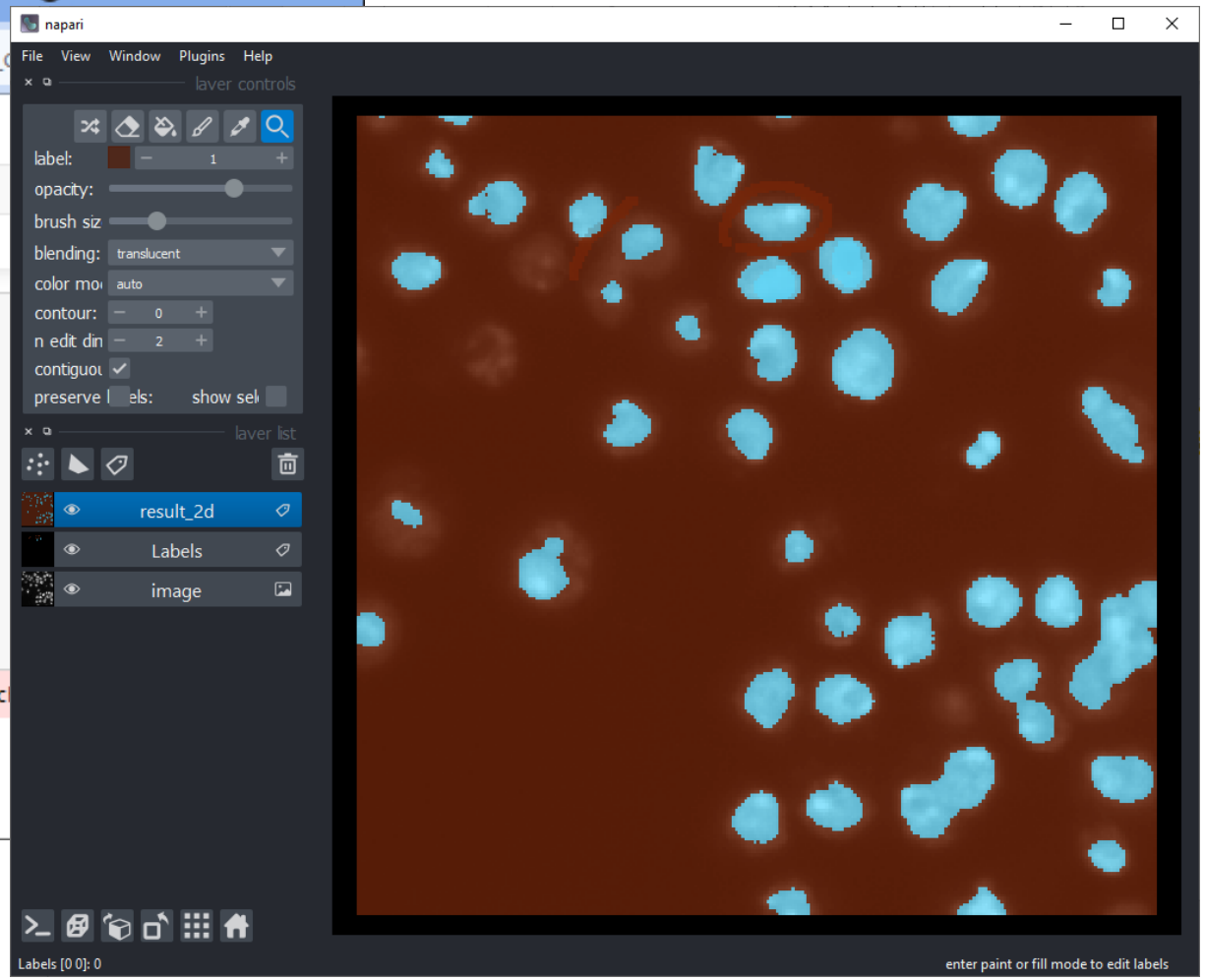


Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Exercise: scikit-learn / napari

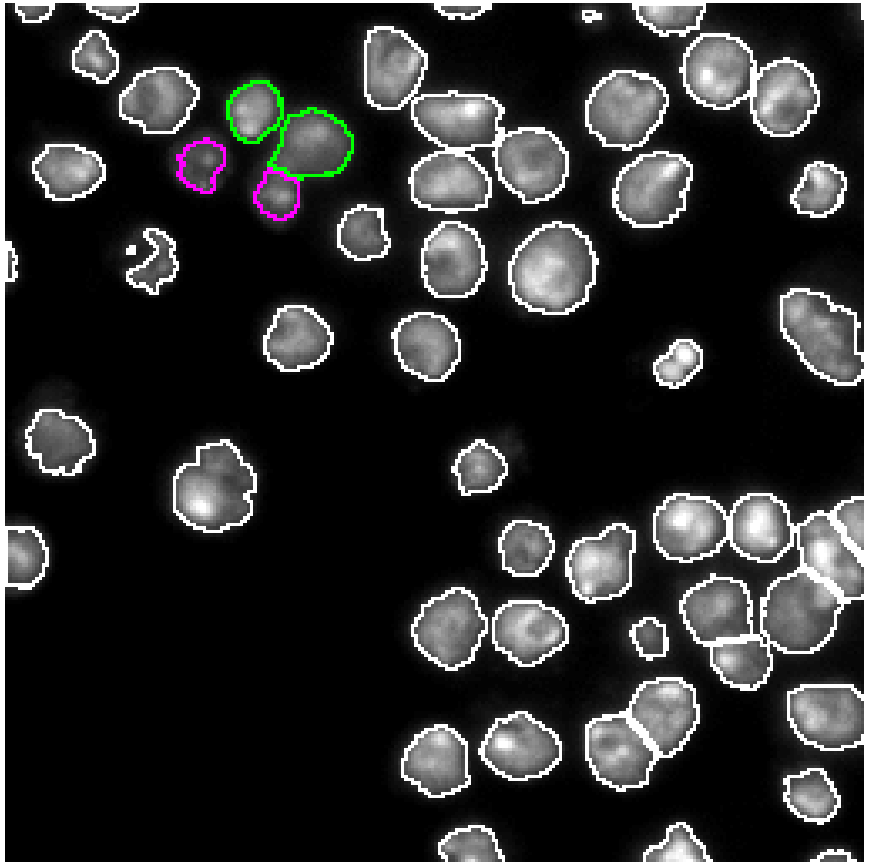
- In the jupyter notebook: Which code changes are necessary to classify three classes of pixels?



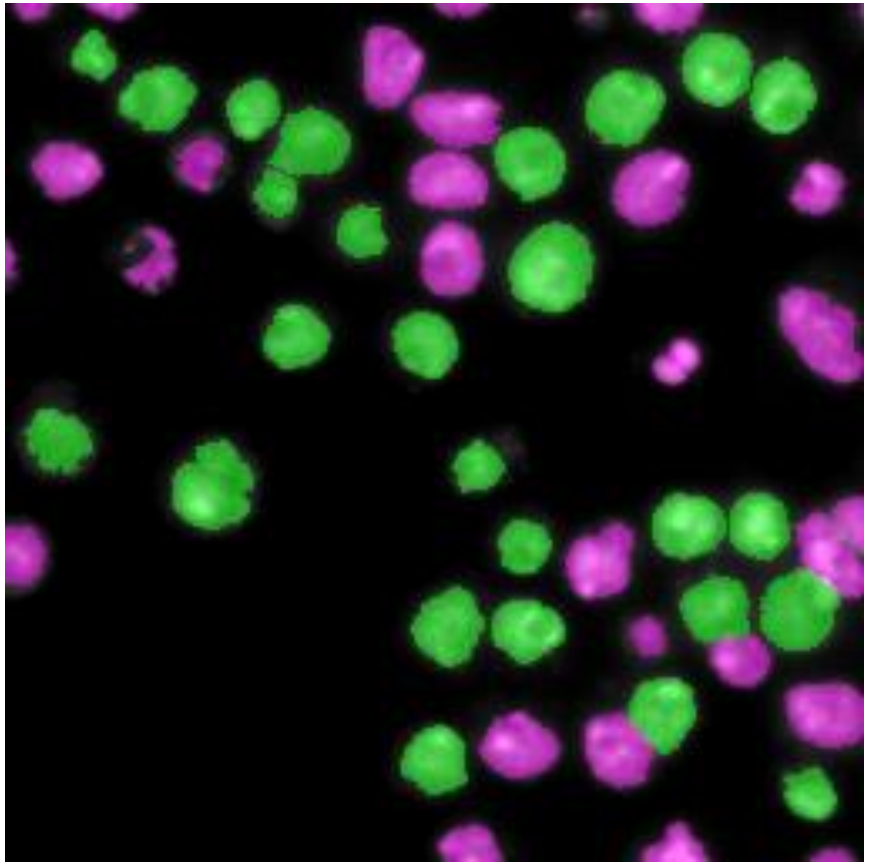


Exercise: Precision / recall

- Calculate precision and recall for the algorithm which created this prediction



Ground truth annotation



Prediction

Summary

Today, you learned

- Machine learning for Pixel and Object classification
- Segmentation / prediction quality measurements
  - Precision
  - Recall
  - Jaccard Index
- Fiji
  - Trainable Weka Segmentation
  - Labkit
- Python
  - Scikit-learn / napari

Coming up next:

- Deep learning
  - Convolutional neural networks