



Project #11: Segmentation and tracking of epithelial cells

This page was created by the [AI4Life project](#) using data provided by Madeline Lovejoy and Rafael Garcia-Mata (PhD) at [Garcia-Mata Lab](#)

If any of the instructions are not working, please [open an issue](#) or contact us at ai4life@fht.org!

Project challenges: denoising, cell instance segmentation, cell tracking.

Table of Contents

1. [Introduction](#)
2. [Installation](#)
3. [Denoising with N2V](#)
4. [Segmentation with Cellpose](#)
5. [Tracking with Trackmate-Cellpose](#)
6. [Conclusion](#)

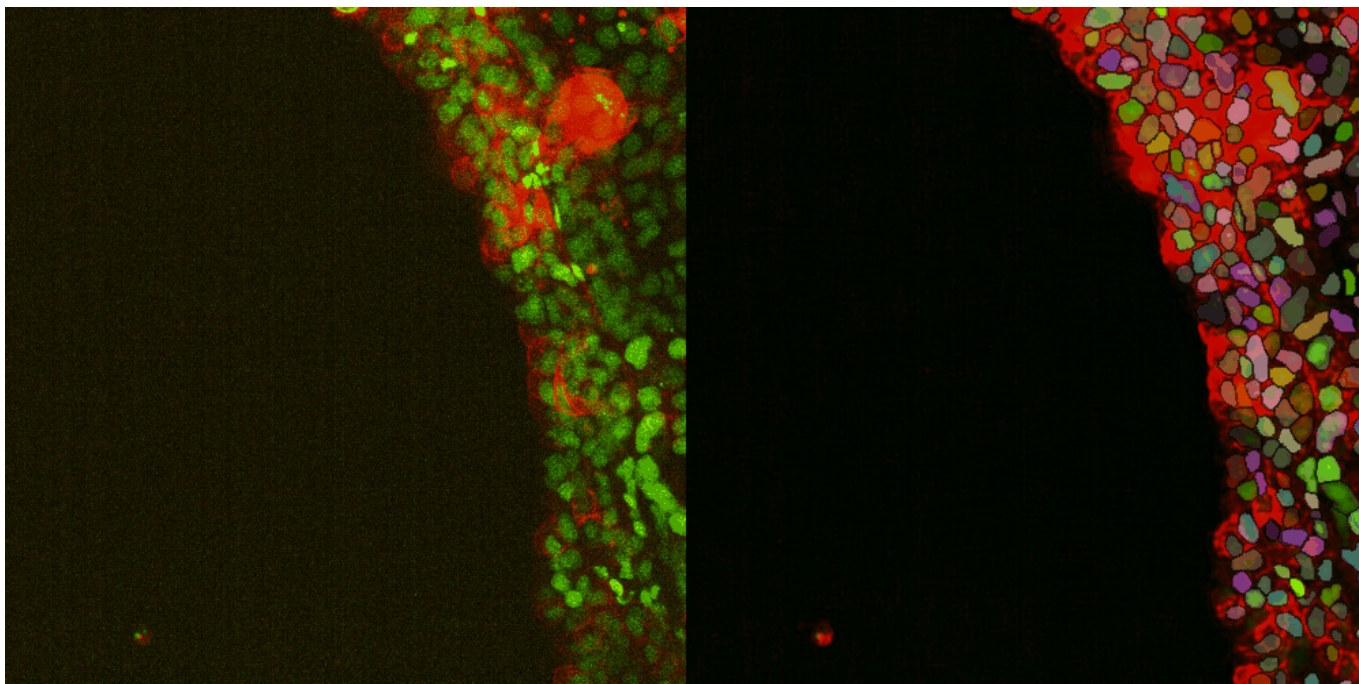
Introduction

Researchers from [The University of Toledo Garcia-Mata Lab](#) are imaging a confluent 2D monolayer of epithelial cells. The monolayer is scratched with a pipette tip, and the video shows the migration of the cells to close the wound. They would like to compare the behavior of different cell lineages automatically. They are interested in segmenting and tracking individual cells and nuclei in time.

In this tutorial, we will demonstrate how to segment and track individual epithelial cells in a series of images obtained with fluorescence microscopy.

We will start by denoising the images using [N2V](#) Python package, which denoises microscopy images using deep learning. Next, we will segment cells using [Cellpose](#), a popular deep-learning cell segmentation algorithm. Finally, we'll track the segmented cells using the TrackMate plugin in Fiji.

Here is a visualization of the original data and resulting tracking:



Data is provided under a CC-BY license.

Let's get started! 🚀

Installation

Install Python package manager

We recommend installing [conda](#) or [mamba](#). Go to the website and follow the installation instructions for your operating system.

During installation, select the installation folder that you have permissions for, such as your home folder.

Install N2V

Follow [N2V installation](#) for official instructions.

► N2V installation details

1. Open terminal (such as Terminal on MacOS, for Windows we recommend [Git for Windows](#))
2. Create and activate a new conda environment:

```
conda create -n n2v python=3.9
conda activate n2v
```

3. Install Tensorflow by following the [official instructions](#). If you have no GPU available, install the CPU version.

N2V only supports Tensorflow 2. We'll now install version 2.13.0 with GPU support as an example:

```
pip install tensorflow[and-cuda]==2.13.0
```

4. Check if the installation is correct. If you are using GPU, this command should return a list of available devices:

```
python -c "import tensorflow as tf;
print(tf.config.list_physical_devices('GPU'))"
```

5. Install N2V and Jupyter:

```
pip install n2v jupyter
```

Install Cellpose

Follow [Cellpose installation](#) for official instructions.

If you are using GPU check out [GPU version \(CUDA\) on Windows or Linux](#) section.

► Cellpose installation details

1. We recommend creating a new separate environment:

```
conda create -n cellpose python=3.9
conda activate cellpose
```

2. Install PyTorch by following the [official instructions](#).

If you have a GPU, select a version that is compatible with your driver version. Follow [this link](#) for older versions.

We will be installing PyTorch version 1.12 and cudatoolkit version 11.3 as an example:

```
conda install pytorch==1.12.0 torchvision==0.13.0 cudatoolkit=11.3 -c
pytorch
```

3. Check if the installation is correct. If you are using GPU, this command should return a list of available devices:

```
python -c "import torch; print([torch.cuda.get_device_properties(i)
for i in range(torch.cuda.device_count())])"
```

4. Install Cellpose:

```
pip install pyqt5 "cellpose[gui]"
```

5. Check the installation by running Cellpose. You should see the Cellpose window open:

```
cellpose
```

Install TrackMate-Cellpose

Follow [TrackMate-Cellpose](#) for official instructions.

► TrackMate-Cellpose installation details

1. Open Fiji.
2. Add The **TrackMate-Cellpose** update site and restart Fiji.
3. In your **cellpose** conda environment, run this command to find where your python is installed:

```
which python
```

4. Open TrackMate and select **Cellpose detector** from the list of available detectors.
5. Insert the path to your python in Path to Cellpose

Step #1: Denoising with N2V

If the images are noisy, denoising can help achieve better segmentation performance. Most deep learning segmentation models are trained with clean images, so noise in the data severely affects the resulting accuracy.

You will likely need to re-train the denoising model for different experiments because differences in noise distribution will affect the performance, but worry not N2V is quite fast to train!

N2V offers a [Python package](#) and a [napari plugin](#), or if you do not have a GPU available, you can try using a Google Colab N2V notebook from [ZeroCostDL4Mic](#).

Usage:

1. Download the example jupyter notebook here: [N2V_denoising.ipynb](#)
2. Download example data series here: **TBD**
3. Activate the conda environment

```
conda activate n2v
```

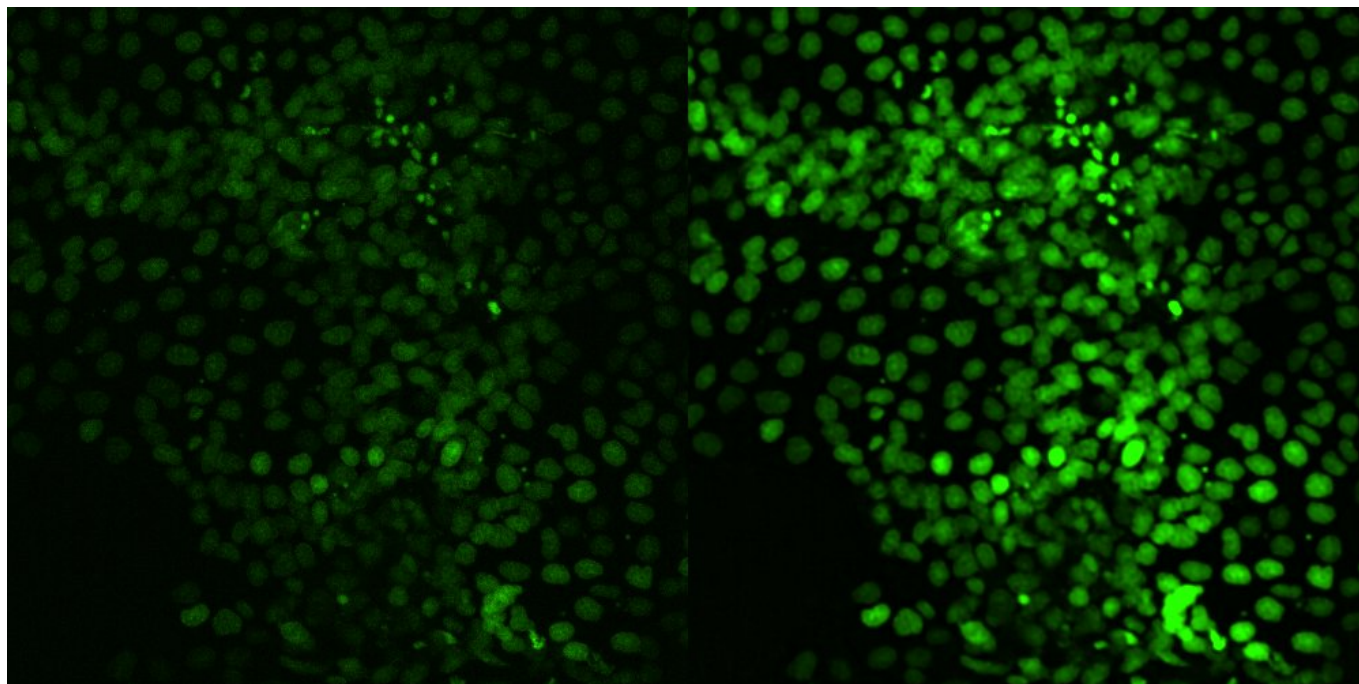
4. Run jupyter service

```
jupyter notebook
```

5. In the opened browser page, open the notebook and follow the instructions.

6. To run a pre-trained model, download and unzip the [pretrained model](#). Then, in the Jupyter Notebook, go to the section 7. *Prediction*.

Here is an example of an original and denoised image:



Step #2: Segmentation with Cellpose

Usage

[Cellpose](#) is a popular deep-learning cell segmentation algorithm.

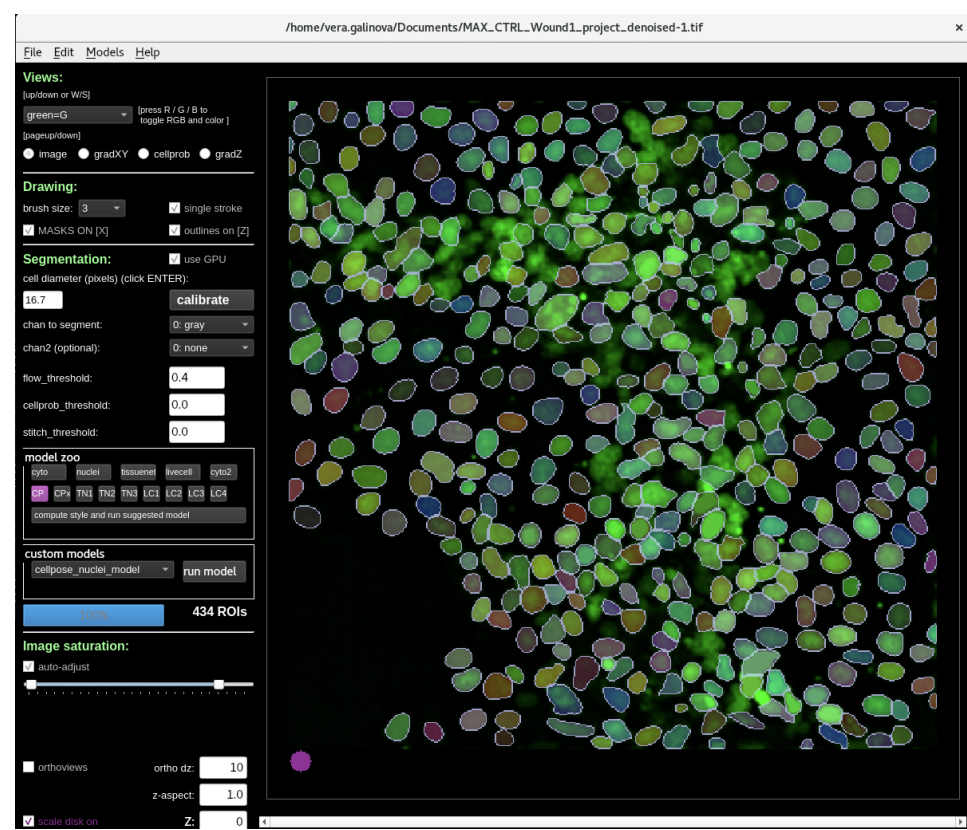
1. Download and unzip pre-trained models for membrane and nuclei segmentation:
[cellpose_membrane_model.zip](#)
2. Activate the conda environment

```
conda activate cellpose
```

3. Run Cellpose

```
cellpose
```

4. Select the channels you want to segment. If you want to segment entire cells, add the membranes as **chan to segment** and nuclei as **chan2**. If you want to segment only nuclei, and them to **chan to segment** and leave **chan2** empty.
5. Calibrate the average object size automatically by clicking **calibrate**. The circle on the bottom-left should be approximately the size of the objects you want to segment.
6. Check how default models perform on the data by clicking any in the **model zoo** section.
7. Add a custom pretrained model to the interface: **Models > Add custom torch model to GUI**.
8. Select this model in the **custom models** section and run it by clicking **run model**.



How to train your own model

You can watch a video tutorial on how to use Cellpose here - <https://www.cellpose.org/>

- 1. Find the best working model from the **model zoo** section.
- 2. Look through the errors in the predictions and correct a few of them. **Click Help > Help with GUI** for help with labeling controls.
- 3. Train a new model on the corrected data. Click **Models > Train new model**.
- 4. Choose the best model from step 3 as an initial model. You cannot select a custom model as an initial model.
- 5. We suggest using the following settings:

train settings

train model w/ images + _seg.npy in current folder >>

filenames # of masks

initial model:

CPx

chan to segment:

green

chan2 (optional):

none

learning_rate

0.001

weight_decay

0.0001

n_epochs

100

model_name

CP_20240122_150538

(to remove files, click cancel then remove from folder and reopen train window)

Cancel

OK

6. After training, you'll see your new model in the **custom models** section. It will also be saved in the folder with your data. **Cell diameter** will also be updated with what the model learned from the data.
7. You can continue iterating: predict with the model - fix a couple of mistakes - train a new model - fix mistakes - train a new model

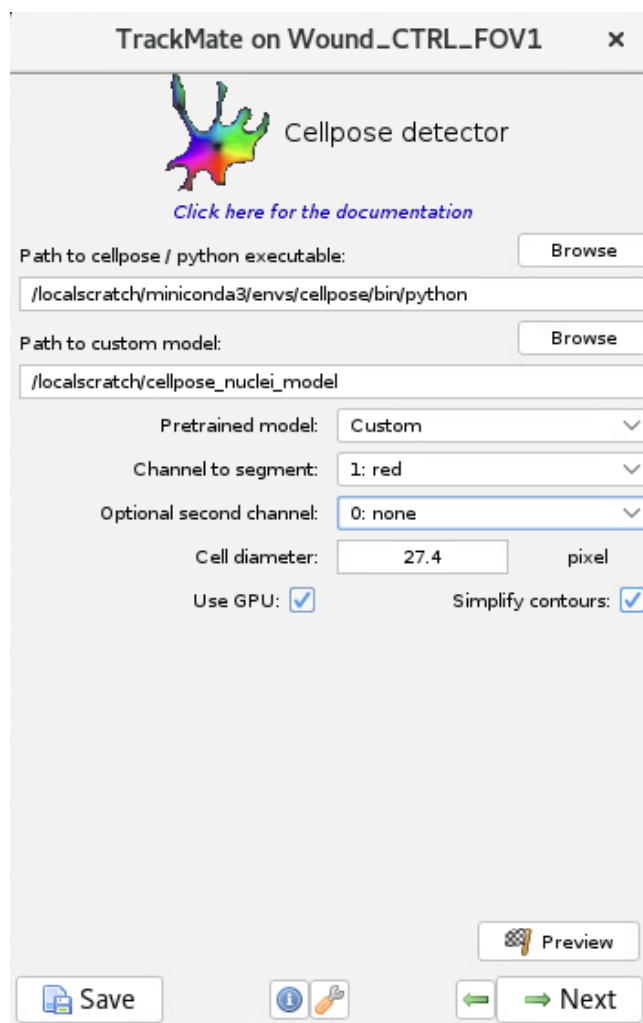
Step #3: Tracking with Trackmate-Cellpose in Fiji

To use the resulting model on a time series and track the segmentations, install the [TrackMate-Cellpose](#) integration in Fiji.

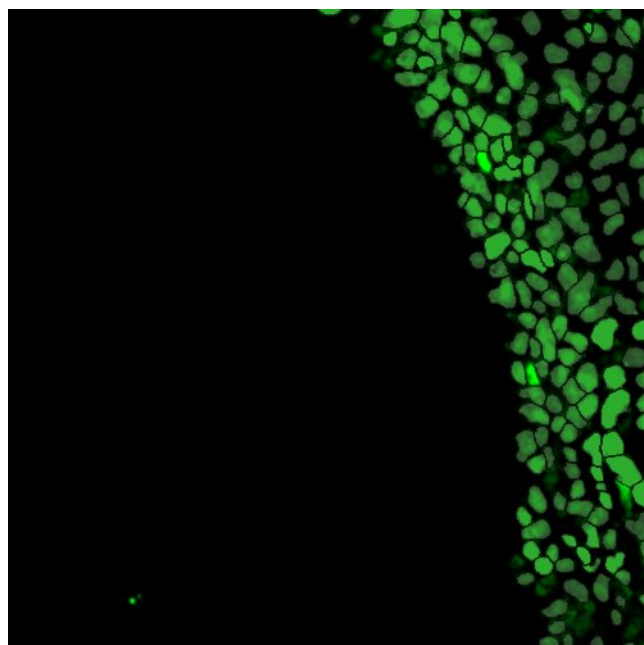
Usage

1. Open a **denoised** time series.
2. Check if **Path to cellpose** is correct.
3. Select **Custom** model from **Pretrained model**.
4. Select the channels you want to segment.
5. Select the path to your model. Enter the channels you want to segment and the approximate diameter of your objects.

Here is an example of the settings:



6. Press **preview** to run prediction on one timepoint and check the result. The objects on the image should now be highlighted.



7. Press **next** to run prediction on all time points (it can take a while). Do not filter the resulting spots.
8. For tracking, we chose the LAP tracker. Here are the settings we used:

Settings for tracker:

LAP Tracker

Frame to frame linking:

Max distance: 30.0 pixel

Feature penalties

Area 1.0

Mean intensity ch1 1.0

+

-

Track segment gap closing:

☒ Allow gap closing

Max distance: 30.0 pixel

Max frame gap: 2

Feature penalties

Area 1.0

Mean intensity ch1 1.0

+

-

Track segment splitting:

☒ Allow track segment splitting

Max distance: 7.0 pixel

Feature penalties

Mean intensity ch1 1.0

+

-

Track segment merging:

☐ Allow track segment merging

Max distance: 15.0 pixel

Feature penalties

+

-

9. Here are the visualization settings we used:

Display options

Edit settings

☒ Display spots ☒ as ROIs

Spot display radius ratio:

Display spot names: ☐

Color spots by:

min max

☒ Display tracks

Fade tracks in time: ☒

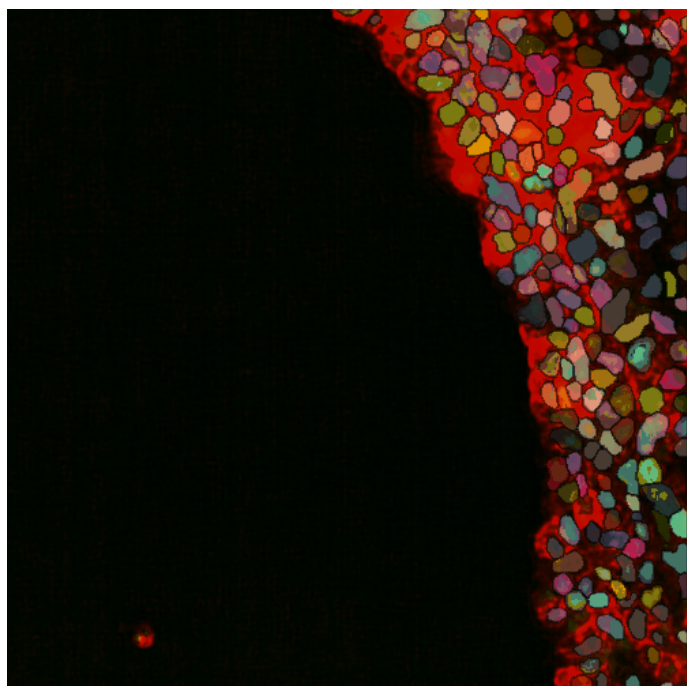
Fade range:

Color tracks by:

min max

☐ Limit drawing Z depth

10. To improve the results, consider different distance settings, feature penalties, and track filtering.



Conclusion

In this tutorial, we showed how to use [N2V](#), <https://www.cellpose.org/>, and [TrackMate-Cellpose](#) to segment and track nuclei of epithelial cells in a series of images obtained with fluorescence microscopy.

[AI4Life](#) is a Horizon Europe-funded project that brings together the computational and life science communities.

AI4Life has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement number 101057970. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.