

# **MyoFuse : How to Use**

## ***Option 1 : Perform nuclei segmentation and classification separately***

This workflow involves three main steps:

1. Perform image pre-processing and nuclei segmentation using the provided Python code (a Jupyter Notebook).
2. Use the Svetlana plugin for Napari to classify nuclei and save prediction images.
3. Extract fusion index values from the images by processing the Svetlana output with another Python code.

This method is particularly recommended for first-time users, as navigating the predictions with the Svetlana plugin helps assess classifier accuracy for your specific needs.

---

## **Requirements**

### **1. Install Jupyter Notebook or JupyterLab**

The easiest way is to install the [Anaconda distribution](#) which includes both tools.

### **2. Install the Svetlana plugin for Napari**

- Ensure you use a Python 3.9 environment.
- Follow the installation instructions provided in the [Svetlana documentation](#).
- Create the environment via Anaconda Navigator or a Python console.
- After installing Napari, update it before proceeding to install Svetlana.

### **3. Install required packages**

- In your environment, run the following command: `pip install -r requirements.txt`
  - Ensure you are in the correct directory before executing this command.
- 

## **Segmentation**

### **1. Prepare your folder structure :**

- Create a folder in the directory of your choice to contain the images to process.
- If your images have both myotube and nuclei fluorescence channels, place them in a subfolder named **“Full Images”**.
- If the images are already split into channels, place them in two separate subfolders named **“Images”** and **“Nuclei”**.
- Add the **“Svetlana”** subfolder containing the following files:
  - `MyoFuse.pth`

## ■ Config.json

### 2. Import packages :

- Open the `MyoFuse.ipynb` notebook in Jupyter Notebook or JupyterLab.
- Run the first cell to import the necessary packages.

### 3. Set up the paths :

```
## ENTER YOUR VALUES ##
```

```
parent_directory = "/path/to/your/folder/" # Replace with the actual path  
cellpose_directory = "/path/to/your/model" # Replace with the actual path
```

```
myotube_channel = 0 # Replace with the actual myotube channel  
nuclei_channel = 1 # Replace with the actual nuclei channel
```

```
dia = 24 # Adjust this value depending on your images. Set it to 0 for Cellpose to automatically determine the best value.
```

```
extension = (".tif", ".tiff")
```

- In the second cell, update the following:
  - `parent_directory`: The directory containing your images and the "Svetlana" folder.
  - `cellpose_directory`: Path to the segmentation model.
- If using multi-channel images, specify the channels corresponding to myotubes and nuclei.
- Adjust the `dia` (diameter) value only if segmentation results are unsatisfying. Set `dia` to 0 for Cellpose to auto-detect the optimal value.

### 4. Perform segmentation :

- Run the second cell.
- If using multi-channel images, execute the third cell to split them.
- Continue executing the notebook cells until the segmentation step is complete.

```
# List all image files in the "Nuclei" folder  
image_files = [f for f in os.listdir(nuclei_folder) if f.lower().endswith(('.tif', '.tiff'))]
```

```
# Process each image  
for file in image_files:  
    file_path = os.path.join(nuclei_folder, file)  
    print(f"Processing file: {file_path}")  
    result = loop(file_path, parent_directory, dia)
```

Segmentation masks will be saved in a new subfolder named **"Masks"**. You can now transition to using Svetlana for classification.

---

## Classification

Perform classification with the Svetlana plugin using the prediction module as explained in the [Svetlana documentation](#). Load the "MyoFuse.pth" model contained in the "Svetlana" subfolder. Alternatively replace the model file by the one that you retrained in the same

folder. Use the batch method to analyze all your images at once and save the results in the “Svetlana” subfolder.

To obtain the nuclei counts and fusion index value, process the “**prediction\_regionprops.xlsx**” file using the notebook “**Fusion Index.ipynb**”. Simply update the path of the “region\_props.xlsx” file at the beginning of the code. The new “**Fusion Index.xlsx**” file containing the results will be generated in the same folder.

```
# Define the directory and file name
directory = 'path/to/your/directory'
filename = 'prediction_regionprops.xlsx'
```

---

### ***Option 2 : Perform nuclei segmentation and classification with a single code***

This method is particularly recommended to improve time efficiency, once the accuracy of the workflow for your specific situation is validated.

Perform all steps of the “**Requirements**” and “**Segmentation**” sections as described in “**Option 1**”. Installation of the Svetlana plugin is not required for this method.

After pre-processing and nuclei segmentation, proceed to classification by running the rest of the notebook. You will need to replace the name of the classifier if you re-trained the model in the following cell :

```
model_path = os.path.join(parent_directory, 'Svetlana', 'MyoFuse.pth')
```

In the root folder of your project, a .xlsx file containing the results will be saved. Predictions images will be saved in the “**Predictions**” subfolder.