Requirements and installation guide for pipeline

1. Hardware requirements (recommended)

This pipeline has been tested for the following system :

- 64-bit Windows 10 Pro
- 2x Intel Xeon Silver 4110 (8 Cores, 3.0GHz) processor (CPU)
- Nvidia Geforce GTX 1080 graphic card (GPU)
- 384 Gb of RAM.

It should also work on other systems, as long as few requirements are met :

- Cellpose, N2V and CLIJ2 processes use GPU. Running pipeline without GPU may not be possible or extremely slow. GPU should be CUDA compatible for Cellpose and N2V (check here : https://developer.nvidia.com/cuda-gpus.)

- Substantial RAM may also be used by Cellpose, we recommend to have > 100Gb of RAM for 3D datasets.

2. Required softwares and plugins

Softwares :

- Anaconda3-2021.11 python distribution : https://www.anaconda.com/products/distribution
- Python 3.7.9
- CUDA toolkit 10.0
- PyTorch 1.6.0
- Cellpose v0.6.1 (Stringer et al., 2021) : See 3.d, https://github.com/MouseLand/cellpose, https://cellpose.readthedocs.io/en/latest/installation.html
- Fiji : https://imagej.net/software/fiji/downloads

Plugins:

- CLAHE (Pizer et al., 1987; Zuiderveld, 1994)
- Progressive intensity and gamma correction (Murtin, 2016)
- Fijiyama (fijiyama-4.0.0) (Fernandez and Moisy, 2021)
- CLIJ2 (clij2-2.5.3.0) (Haase et al., 2020)
- MorpholibJ (morpholibJ-1.4.3) (Legland et al., 2016)
- Noise2Void (n2v-0.8.6) (Krull et al., 2019)
- CSBDeep (csbdeep-0.6.0) (Weigert et al., 2018)

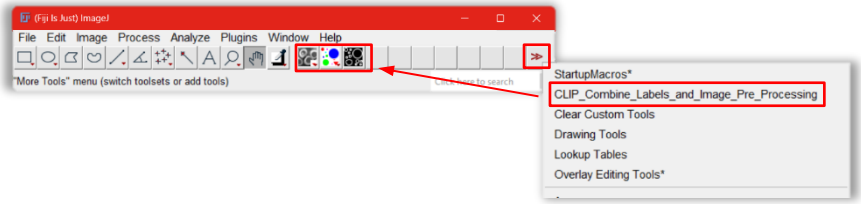
3. Installation guide

a. CLIP macro tool installation

Place the file *CLIP_Combine_Labels_and_Image_Pre_Processing.ijm* in toolsets folder of Fiji : ... > *Fiji.app* > *macros* > *toolsets*

Place images *image.png*, *Combine.png* and *Label_BND.png* in *icons* folder (or create one) : ...> *Fiji.app* > *macros* > *toolsets* > *icons*

Restart Fiji. Select *more tools* and click on *CLIP_Combine_Labels_and_Image_Pre_Processing*. Three tool menus will appear.



b. Plugins Fiji for image preparation (manual)

- **Intensity enhancement :** The code is available here.

A file *Progressive_Intensity_and_Gamma_Correction.class* is provided on our GitHub (https://github.com/INRAE-LPGP/ImageAnalysis_CombineLabels).
Place the file *Progressive_Intensity_and_Gamma_Correction.class* in Fiji plugins repertory : ...> *Fiji.app* > *plugins.*
Restart Fiji. The plugin will be available in : menu *Plugins › Progressive Intensity and Gamma Correction.*

- **Contrast enhancement :** Uses ***Enhance Local Contrast (CLAHE)*** Fiji plugin. It is available in our CLIP macro tool in the CLIP_Image menu (*See 3.a for installation*)

- **3D registration :** Uses ***Fijiyama*** Fiji plugin. Installation information are available here : https://imagej.net/plugins/fijiyama#installation

c. Plugins Fiji for CLIP macro (automatic pre- and post-processing)

- **CLIJ2 :** Enables GPU access for various macros and functions on Fiji. Installation information are available here : https://clij.github.io/clij2-docs/installationInFiji

- **MorpholibJ :** Collection of mathematical morphology methods for Fiji/ImageJ. Installation information are available here : https://imagej.net/plugins/morpholibj#installation

- **CSBDeep / N2V :** Uses neural networks for image denoising in Fiji. Installation information are available here : https://imagej.net/plugins/n2v#installation. For GPU support, see documentation here : https://imagej.net/develop/tensorflow

d. Setup for Cellpose

Help for Cellpose environment installation and activation is provided below. For further details about Cellpose installation, please refer directly to Authors documentation (Stringer et *al*) :
https://github.com/MouseLand/cellpose
https://cellpose.readthedocs.io/en/latest/installation.html

---------- **Installation** ----------

First install **Anaconda distribution** (Python 3.8, https://www.anaconda.com/products/distribution, https://repo.anaconda.com/archive/), don't add anaconda to the path

- Add a new path to Anaconda prompt

Find shortcut "anaconda prompt", right click, properties :

- In "Start in" paste the path to the folder "cellpose-master" downloaded from Github
- Open Anaconda prompt, you should have :

```
(base) D:\Your_Path\cellpose-master>
```

- Change Python version to 3.7 for Cellpose v.0.6.1 :
  - Check available versions :

```
(base) D:\Your_Path\cellpose-master> conda search python
```

  - Select version 3.7.9 :

```
(base) D:\Your_Path\cellpose-master> conda install python=3.7.9
```

- Create environment :

```
(base) D:\Your_Path\cellpose-master> conda env create –f environment.yml
```

- Activate Cellpose (needed at each launch) :

```
(base) D:\Your_Path\cellpose-master>
(base) D:\Your_Path\cellpose-master> conda activate cellpose
(cellpose) D:\Your_Path\cellpose-master>
```

- Upgrade cellpose (No-GPU version) :

Download desired version https://github.com/MouseLand/cellpose/releases
Copy file "cellpose-0.6.1.tar.gz" (*.tar.gz) in the folder "cellpose-master", then run :

```
(cellpose) D:\Your_Path\cellpose-master> pip install cellpose --upgrade
```

---------- **Installation with GPU** ----------

- Check GPU version :

```
(cellpose) D:\Your_Path\cellpose-master> nvcc --version
```

Or :

```
(cellpose) D:\Your_Path\cellpose-master> nvcc -V
```

- Check CUDA version supported by GPU and running processes :

```
(cellpose) D:\Your_Path\cellpose-master> nvidia-smi
```

- Uninstall torch (CPU version) + mxnet

```
> pip uninstall torch
> pip uninstall mxnet-mkl
```

  - For CUDA version 10.0 (+CPU version accelerated when GPU not used) (*not used here*):

```
> pip install mxnet-cu100mkl
```

  - For CUDA version 10.0 (+CPU not accelerated) (*not used here*):

```
> pip install mxnet-cu100
```

- Install Cuda toolkit and pytorch (you can specify pytorch version (pytorch==1.6.0 cudatoolkit….))

```
> conda install –c anaconda cudatoolkit
```

```
> conda install pytorch cudatoolkit=10.0 –c pytorch
```

- For futur upgrades (to avoid installation of CPU version and torch) :

```
> pip install --no deps cellpose --upgrade
```

---------- **Troubleshooting** ----------

- Change parameters not available in --args (depending on Cellpose version)

In script « models.py » : D:\Your_Path\cellpose-master\cellpose\models.py (**Line 120)**

```
def eval   (self, x, batch_size=4, channels=None, invert=False, normalize=True,
diameter=30, do_3D=True, anisotropy=2.2, net_avg=True, augment=False, tile=True,
tile_overlap=0.1, resample=True, interp=True, flow_threshold=0.6,
cellprob_threshold=-1, min_size=10, stitch_threshold=0.0, rescale=None,
progress=None)
```

- `OverFlowError : cannot serialize a bytes object larger than 4GB`

In script «format.py » (Numpy library) : D:\Your_Path\anaconda3\envs\cellpose\Lib\site-packages\numpy\lib (**Line 664 (in write_array))**

```
pickle.dump(array, fp, protocol=3, **pickle-kwargs)
```

Replace protocol=3 by **protocol=4**