

EMITO-METRIX APPLICATION: INSTALLATION TUTORIAL

EMito-Metrix is a high-performance pipeline for analyzing mitochondrial morphology and ultrastructure in multiple tissues from low to high-resolution images acquired with Electron microscopy (EM). The application allows to compute a set of morphometrics and texture measurements, and provides a list of graphs for optimizing data visualization using dimensionality reduction (UMAP or PCA) and more conventional depiction of data distribution (density curves, histograms, violin plots or star plots). Additionally, a machine learning (ML) module with predictive analytic tools allows to determine how any given experimental condition would impact on mitochondrial morphology and ultrastructure

EMito-Metrix application was written by Mathieu Vigneau, Emmanuel Doumard and Jean-Philippe Pradère from the RESTORE Institute.




This section describes installation instructions for Windows, Linux and MacOS operating systems

- SYSTEM REQUIREMENTS

Windows, Linux and MacOS are supported for running the EMito_Metrix application. Before installing, we recommend to check the following requirements for each software and environment used by the application:

- Anaconda distribution requirements:
<https://docs.anaconda.com/anaconda/install/>
- Fiji software requirements:
<https://imagej.net/software/fiji/downloads#system-requirements>
- Cellpose software requirements:
<https://github.com/MouseLand/cellpose?tab=readme-ov-file#system-requirements>

- TABLE OF CONTENTS

• Emito-Metrix installation for Windows.....	
• Emito-Metrix installation for MacOS.....	
• Emito-Metrix installation for Linux.....	

EMITO-METRIX INSTALLATION FOR WINDOWS

1- CELLOSE PYTHON ENVIRONMENT: INSTALLATION AND SETTING

- Anaconda distribution installation

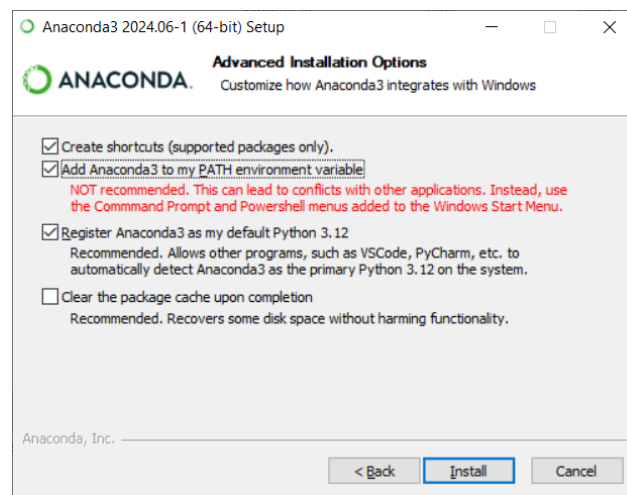
Install an Anaconda distribution for Python 3 using the following URL:

<https://www.anaconda.com/download/success>

Download the latest version of Anaconda installer for Windows 64bits. After downloading, go to your download folder and double-click the installer to launch Anaconda installation.

It is recommended that you install Anaconda using the *Just Me* option, which will install Anaconda Distribution to just the current user account (doesn't require Windows Administrator privileges).

During installation, add Anaconda to your PATH environment variable, as shown below:



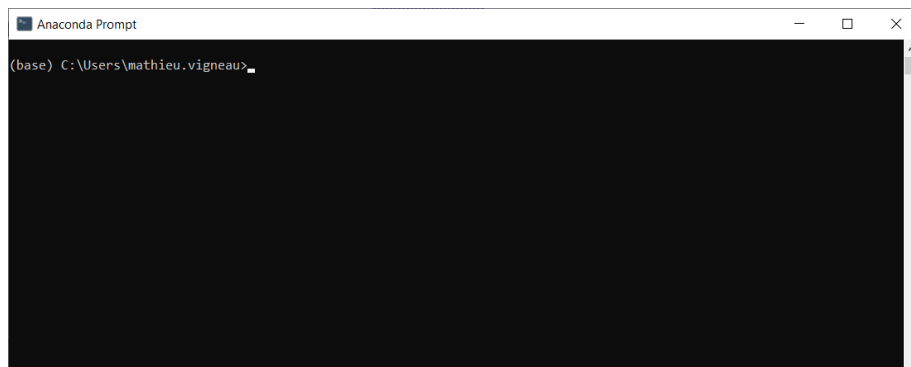
See [here](#) for detailed instructions of Anaconda installation and settings

- Cellpose installation & setting

Once you installed anaconda environment, you will need to create and set a python virtual environment for Cellpose application (which is essential for the EMito-Metrix application).

- **Open an anaconda prompt**

From the Windows start menu (🔍), type *anaconda prompt* in the search bar and click on the Anaconda application to start it. This will open a new command prompt window with Anaconda enabled, as shown below:



- **Create the Cellpose Python environment**

We are going to create a **Python virtual environment (venv) for Cellpose** application.

A virtual environment is simply a Python subdirectory with its own file structure, python packages and settings. This Virtual environment let you have a stable and reproducible Cellpose environment, which is essential for the proper running of the EMito-Metrix application.

To create the Cellpose venv, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda create --name cellpose python=3.9
```

This command line **will create a new python3.9 environment named *cellpose***.

This will also create a new folder, containing your Cellpose python environment. To get the path of your cellpose environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda info --envs
```

That will list the paths of all you environments, as shown below:

```
Anaconda Prompt

(base) C:\Users\mathieu.vigneau>conda info --envs
# conda environments:
#
base                * C:\Users\mathieu.vigneau\AppData\Local\anaconda3
cellpose            C:\Users\mathieu.vigneau\AppData\Local\anaconda3\envs\cellpose
emitomatrix         C:\Users\mathieu.vigneau\AppData\Local\anaconda3\envs\emitomatrix
                    D:\Pro\conda-envs\cellpose
                    D:\Pro\conda-envs\cellpose2
                    D:\Pro\conda-envs\emitomatrix

(base) C:\Users\mathieu.vigneau>
```

The cellpose-environment folder is mostly found under the following Windows path:

C:\Users\username\AppData\Local\anaconda3\envs\cellpose or

C:\Users\username\AppData\anaconda\envs\cellpose

- **Cellpose environment activation**

To activate your Cellpose python environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda activate cellpose
```

```
Anaconda Prompt - "C:\ProgramData\anaconda3\condabin\conda.bat" activate cellpose

vc                pkgs/main/win-64::vc-14.2-h2eaa2aa_1
vs2015_runtime    pkgs/main/win-64::vs2015_runtime-14.29.30133-h43f2093_3
wheel             pkgs/main/win-64::wheel-0.43.0-py311haa95532_0
xz               pkgs/main/win-64::xz-5.4.6-h8cc25b3_1
zlib             pkgs/main/win-64::zlib-1.2.13-h8cc25b3_1

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate cellpose
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\mathieu.vigneau>conda create --name cellpose python=3.11
(base) C:\Users\mathieu.vigneau>conda activate cellpose
(cellpose) C:\Users\mathieu.vigneau>
```

- **Cellpose package installation**

Next, you need to install Cellpose application from your Cellpose python venv. In your opened anaconda prompt, type the following python command line, and press Enter:

```
python -m pip install cellpose[gui]
```

This command line will install Cellpose python packages in your python virtual environment, also including the **Cellpose Graphical user Interface**.

- **(Optional) Cellpose GPU package installation**

If you have a **CUDA supported GPU** installed on your computer (see [here](#) for CUDA GPU compatibility), you can install a GPU version of Cellpose.

Please see this [URL](#) for the install instructions of CUDA toolkit and GPU version of Cellpose.

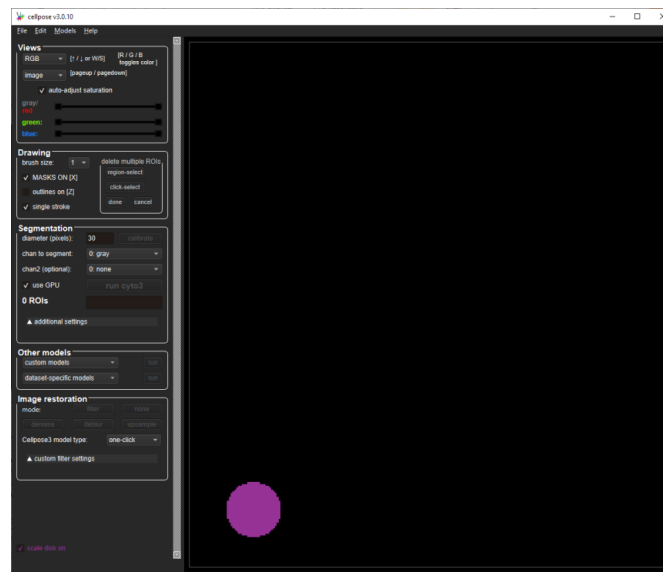
If you need more information about Cellpose detailed requirements, dependencies and installation instructions, see [here](#).

- **(Optional) Cellpose package checking**

Once you installed the Cellpose package, we recommend verifying the validity of the installation. From your Cellpose python environment, type the following python command lines and press Enter:

```
cellpose
```

If the package installation is correct, the following window will appear:



If you have issues with installation, see the cellpose [docs](#) for more details.

- **Cellpose environment deactivation**

Once the Cellpose installation is correct, **deactivate your Cellpose python environment** typing in your opened anaconda prompt the following python command line:

```
conda deactivate
```

2- FIJI INSTALLATION AND SETTING

- **Fiji software installation**

Install Fiji software distribution using this following URL:

<https://imagej.net/software/fiji/downloads>

Download the latest version of Fiji for Windows 64-bit or Windows-32-bits using the *imagej.net* (USA) link. After downloading, go to your download folder and double-click the *Fiji-zip* file to uncompress it.

It is highly recommended that you store your Fiji app directory in your user space (C:\Users\username\MyDocuments\FijiApp) rather than in the Program Files windows-system directory (write permission issues in these system-wide folders could prevent Fiji updating)

See [here](#) for detailed installation instructions and system requirements of Fiji application

- FIJI plugin updating & setting

EMito-Metrix plugin relies on the following Fiji plugins, which are not automatically installed during the Fiji software installation:

- Bio-Formats (<https://imagej.net/formats/bio-formats>)

The Bio-Formats Importer is a plugin for reading data into Fiji. It can open many dozens of proprietary life sciences formats

- ImageScience (<https://imagej.net/libs/imagescience>)

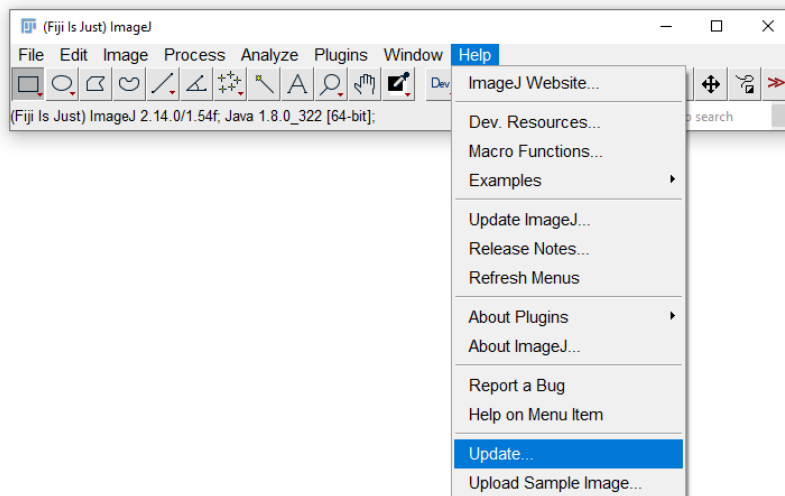
ImageScience is a library of Java classes to facilitate scientific image computing.

- PTBIOP (<https://wiki-biop.epfl.ch/en/ipa/fiji/update-site>)

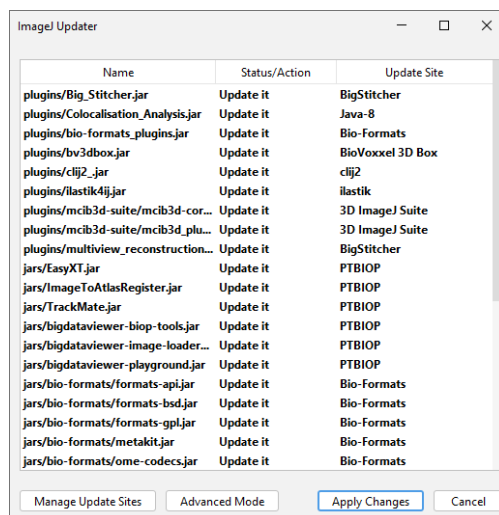
The EPFL - PTBIOP (BioImaging & Optics Platform) is a Fiji package including plugins, macros, ActionBars and Wrappers for external software calls (Cellpose, Elastix, Ilastix...)

- IBMP CNRS

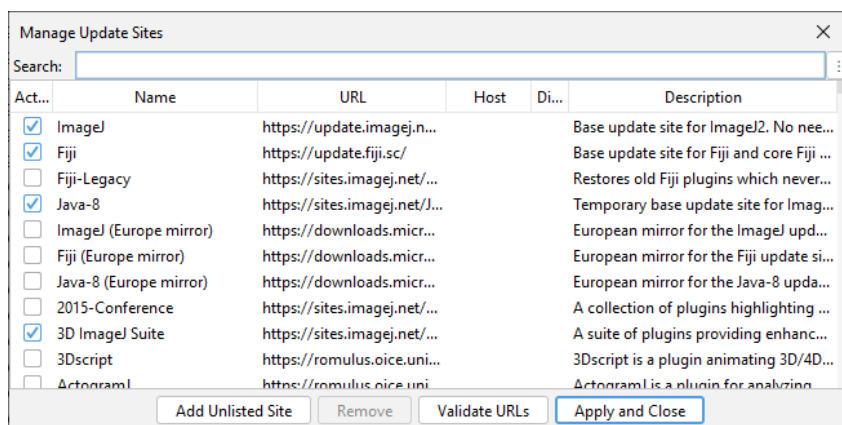
- In your Fiji App directory, double-click the *ImageJ-win64.exe* file to start Fiji software
- Once the software is opened, select *update* from the *Fiji Help menu* (as shown below) in order to update Fiji plugins



- In the *ImageJ Updater* window, select the *Manage update sites* button (see below).



- The *Manage Update Sites* window appears, which contains a list of Fiji plugins you can upload and install in your Fiji application (plugins already installed are checked).



From this *Manage Update Sites* window, check the following plugins to install it:

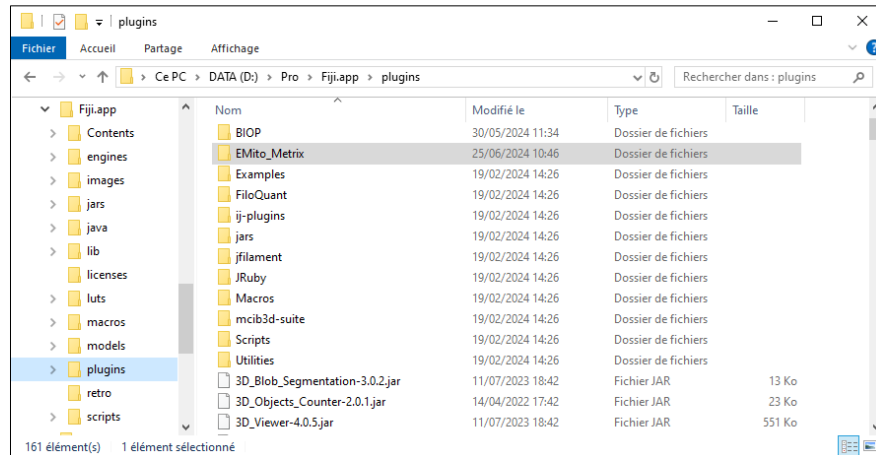
- Bio-Formats
- IBMP-CNRS
- ImageScience
- PTBIOP

- Once these plugins are checked, click on *Apply and Close* button from the *Manage Update Sites* window. Then, click on *Apply Changes* button from the *ImageJ Updater* window
- Once the plugin are installed, restart Fiji application

- **EMito-Metrix plugin installation & setting**

- **EMito Metrix folder creation**

In your Fiji plugins folder (*yourFijiFolder\plugins*), create a new folder named **EMito_Metrix**



- **EMito Metrix file upload**

Download and save the EMito_Metrix files in your *EMito_Metrix Fiji folder*, using the following GitHub repository:

<https://github.com/Restore-Lab/EMito-Metrix>

You must upload these following files from the GitHub repository (!! Keep the same file names !!) :

- EMito_Metrix_Fiji_.ijm (Fiji plugin file)
- dataComputation.py (python file for data display)
- mlComputation.py (python file for data computation and prediction)
- requirements.txt (python package dependencies)
- GeneralistModel_GM_EM.zip (Cellpose GM model)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_FLY_EM.zip (Cellpose SM model for Fly)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_HUMAN_EM.zip (Cellpose SM model for Human)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_MOUSE_EM.zip (Cellpose SM model for Mouse)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_ZEBRAFISH_EM.zip (Cellpose SM model for Zebrafish)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

3- EMITOMETRIX PYTHON ENVIRONMENT: INSTALLATION & SETTING

- **Create the Emitometrix Python environment**

We are going to create a **Python virtual environment (venv)** for data computation plugins.

A virtual environment is simply a Python subdirectory with its own file structure, python packages and settings. This Virtual environment let you have a stable and reproducible environment for data visualization and computation plugins, which are essential for the proper running of the EMito-Metrix application.

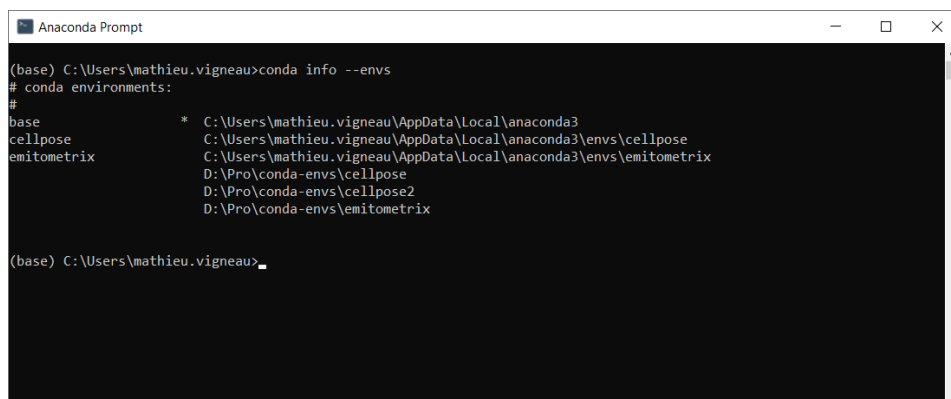
To create the emitometrix venv, type the following python command line in your anaconda prompt, and press Enter:

```
conda create --name emitometrix python=3.11
```

This command line **will create a new python3.11 environment named *emitometrix***. To get the path of your emitometrix environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda info --envs
```

That will list the paths of all you environments, as shown below:



```
Anaconda Prompt
(base) C:\Users\mathieu.vigneau>conda info --envs
# conda environments:
#
base                * C:\Users\mathieu.vigneau\AppData\Local\anaconda3
cellpose            C:\Users\mathieu.vigneau\AppData\Local\anaconda3\envs\cellpose
emitometrix         C:\Users\mathieu.vigneau\AppData\Local\anaconda3\envs\emitometrix
                    D:\Pro\conda-envs\cellpose
                    D:\Pro\conda-envs\cellpose2
                    D:\Pro\conda-envs\emitometrix

(base) C:\Users\mathieu.vigneau>
```

The emitometrix-environment folder is mostly found under the following Windows path:

C:\Users\username\AppData\Local\anaconda3\envs\emitometrix or

C:\Users\username\AppData\anaconda\envs\emitometrix

- **Emitometrix environment activation**

To activate your **emitometrix python environment**, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda activate emitometrix
```

- **Python dependencies installation**

EMito-Metrix plugin relies on the following Python packages, which are not automatically installed during emitometrix python environment installation:

- cloudpickle==3.0.0	- matplotlib==3.9.0	- pyparsing==3.1.2	- six==1.16.0
- contourpy==1.2.1	- numba==0.60.0	- python-dateutil==2.9.0.post0	- slicer==0.0.8
- cycler==0.12.1	- numpy==1.26.4	- pytz==2024.1	- threadpoolctl==3.5.0
- fonttools==4.53.0	- packaging==24.1	- scikit-learn==1.5.1	- tqdm==4.66.4
- joblib==1.4.2	- pandas==2.2.2	- scipy==1.14.0	- tzdata==2024.1
- kiwisolver==1.4.5	- pillow==10.4.0	- seaborn==0.13.2	- umap-learn==0.5.6
- llvmlite==0.43.0	- pynndescent==0.5.13	- shap==0.46.0	- xgboost==2.1.0

The list of these packages is included in the *requirements.txt* file that is found under your *EMito_Metrix Fiji* folder

example of EMitometrix folder : C:\Users\username\Documents\Fiji.app\plugins\EMito_Metrix

To install python dependencies in your emitometrix environment, type the following python command line from your opened anaconda prompt, and press Enter:

```
pip install -r yourFijiFolder\plugins\EMito_Metrix\requirements.txt
```

- **Cellpose environment deactivation**

Once the Cellpose installation is correct, **deactivate your Emitometrix python environment** typing in your opened anaconda prompt the following python command line:

```
conda deactivate
```

Once you correctly installed these software and environments, the EMito-Metrix application is ready for running. See the *EMitoMetrix tutorial* for running instructions.

EMITO-METRIX INSTALLATION FOR MACOS

1- CELLOSE PYTHON ENVIRONMENT: INSTALLATION AND SETTING

- Anaconda distribution installation

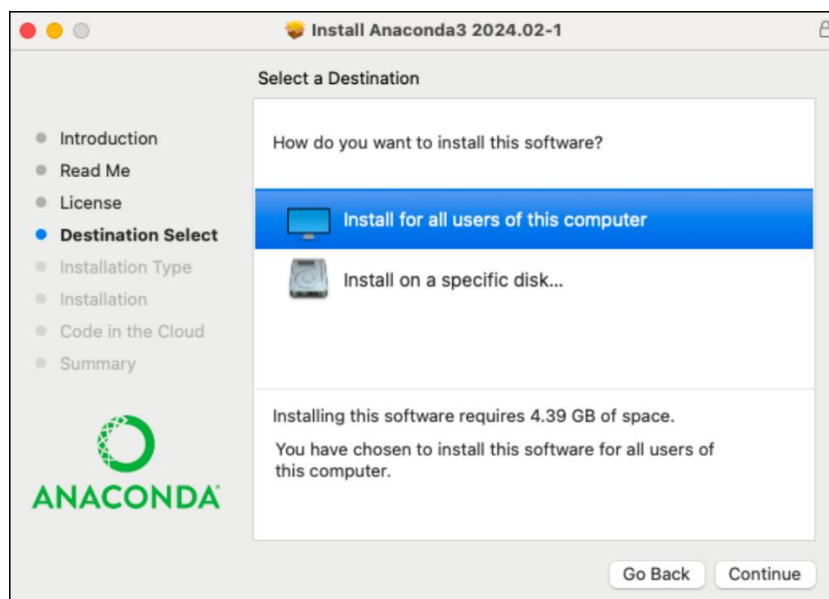
Install an Anaconda distribution for Python 3 using the following URL:

<https://www.anaconda.com/download/success>

Download the 64-bits graphical installer (.pkg) for your system. After downloading, go to your download folder and double-click the installer to launch Anaconda installation.

Double-click the downloaded file and click Continue to start the installation. Answer the prompts on the Introduction, Read Me, and License screens.

Anaconda recommends that you choose Install for all users of this computer (as shown below). As of version 2024.02-1, the default location of the installer is `/opt/anaconda3`*. To install elsewhere, select Install on a specific disk....



* This folder will be used for the following steps

See [here](#) for detailed instructions of Anaconda installation and settings

- **Cellpose installation & setting**

Once you installed anaconda environment, you will need to create and set a python virtual environment for Cellpose application (which is essential for the EMito-Metrix application).

- **Open an anaconda prompt**

From the Finder menu, type *Terminal* in the search bar and click on the *Terminal* icon to start it. This will open a new command prompt window with Anaconda enabled.

- **Create the Cellpose Python environment**

We are going to create a **Python virtual environment (venv) for Cellpose** application.

A virtual environment is simply a Python subdirectory with its own file structure, python packages and settings. This Virtual environment let you have a stable and reproducible Cellpose environment, which is essential for the proper running of the EMito-Metrix application.

To create the Cellpose venv, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda create --name cellpose python=3.9
```

This command line **will create a new python3.9 environment named *cellpose***.

This will also create a new folder, containing your Cellpose python environment. To get the path of your cellpose environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda info --envs
```

That will list the paths of all you environments. This cellpose-environment folder is mostly found under the following Windows path*:

/opt/anaconda3/envs/cellpose

*If *opt* folder is hidden, open *Finder* and select *Macintosh HD* on the left side. Press *Command + Shift + .*

- **Cellpose environment activation**

To activate your Cellpose python environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda activate cellpose
```

- **Cellpose package installation**

Next, you need to install Cellpose application from your Cellpose python venv. In your opened anaconda prompt, type the following python command line, and press Enter:

```
python -m pip install cellpose[gui]
```

This command line will install Cellpose python packages in your python virtual environment, also including the **Cellpose Graphical user Interface**.

- **(Optional) Cellpose GPU package installation**

If you have a **CUDA supported GPU** installed on your computer (see [here](#) for CUDA GPU compatibility), you can install a GPU version of Cellpose.

Please see this [URL](#) for the install instructions of CUDA toolkit and GPU version of Cellpose.

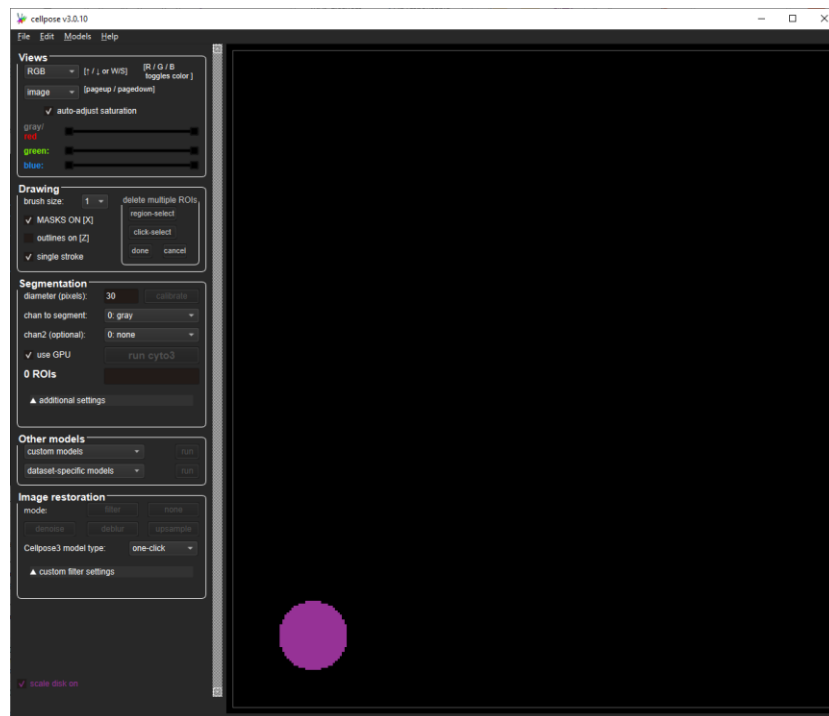
If you need more information about Cellpose detailed requirements, dependencies and installation instructions, see [here](#).

- **(Optional) Cellpose package checking**

Once you installed the Cellpose package, we recommend verifying the validity of the installation. From your Cellpose python environment, type the following python command lines and press Enter:

```
cellpose
```


If the package installation is correct, the following window will appear:



If you have issues with installation, see the cellpose [docs](#) for more details.

- **Cellpose environment deactivation**

Once the Cellpose installation is correct, **deactivate your Cellpose python environment** typing in your opened anaconda prompt the following python command line:

```
conda deactivate
```

2- FIJI INSTALLATION AND SETTING

- Fiji software installation

Install Fiji software distribution using this following URL:

<https://imagej.net/software/fiji/downloads>

Download the latest version of Fiji for macOS (x86_64) using the *imagej.net (USA)* link. After downloading, go to your download folder and double-click the *Fiji.zip* file to uncompress it. Once the application is installed, drag and drop your *Fiji.app* into the application folder.

See [here](#) for detailed installation instructions and system requirements of Fiji application

- FIJI plugin updating & setting

EMito-Metrix plugin relies on the following Fiji plugins, which are not automatically installed during the Fiji software installation:

- Bio-Formats (<https://sites.imagej.net/Bio-Formats/>)

The Bio-Formats Importer is a plugin for reading data into Fiji. It can open many dozens of proprietary life sciences formats

- ImageScience (<https://sites.imagej.net/ImageScience/>)

ImageScience is a library of Java classes to facilitate scientific image computing.

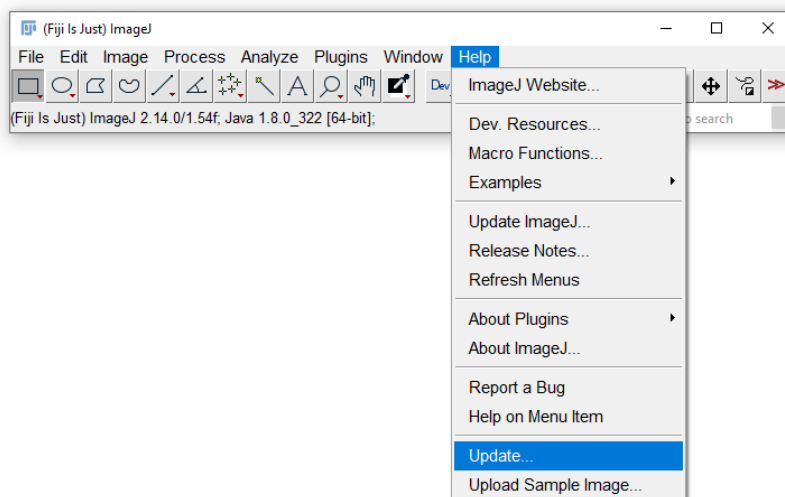
- PTBIOP (<https://wiki-biop.epfl.ch/en/ipa/fiji/update-site>) (<https://biop.epfl.ch/Fiji-Update>)

The EPFL - PTBIOP (BioImaging & Optics Platform) is a Fiji package including plugins, macros, ActionBars and Wrappers for external software calls (Cellpose, Elastix, Ilastix...)

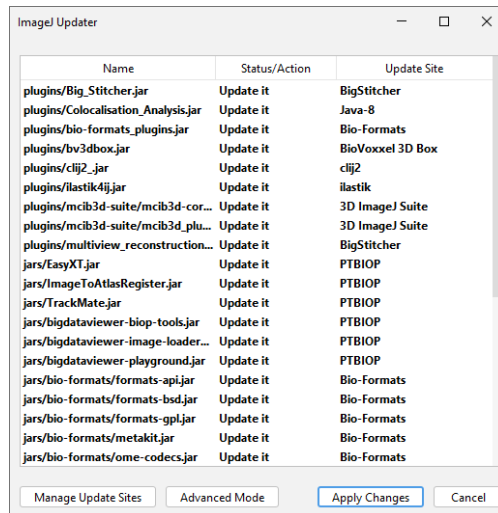
- IBMP CNRS (<https://sites.imagej.net/Mutterer/>)

- 3D ImageJ Suite (<https://sites.imagej.net/Tboudier/>)

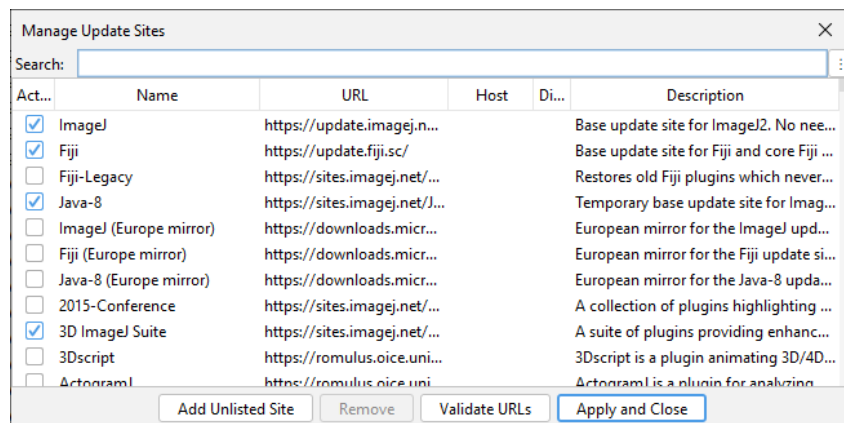
- In your *Fiji App* directory, right-click on the *Fiji.app* file and select *open* to the start Fiji software.
- Once the software is opened, select *update* from the *Fiji Help menu* (as shown below) in order to update Fiji plugins



- In the *ImageJ Updater* window, select the *Manage update sites* button (see below).



- The *Manage Update Sites* window appears, which contains a list of Fiji plugins you can upload and install in your Fiji application (plugins already installed are checked).



From this *Manage Update Sites* window, check the following plugins to install it:

- Bio-Formats
- IBMP-CNRS
- ImageScience
- PTBIOP

If one or more plugins are not available in the *Manage Update Sites* window, you can add the sites manually using the *Add Unlisted Site* button. For each unlisted plugin, specify the corresponding URL from this [list](#). Once all plugins are manually specified, check it.

- Once these plugins are checked, click on *Apply and Close* button from the *Manage Update Sites* window. Then, click on *Apply Changes* button from the *ImageJ Updater* window
- Once the plugin are installed, restart Fiji application

- EMito-Metrix plugin installation & setting

- **EMito_Metrix folder creation**

In your *Application* folder, right-click on the *Fiji.app* icon and select *Show Package Contents*. In your Fiji plugins folder (*yourFijiFolder\plugins*), create a new folder named **EMito_Metrix**

- **EMito_Metrix file upload**

Download and save the EMito_Metrix files in your *EMito_Metrix Fiji folder*, using the following GitHub repository:

<https://github.com/Restore-Lab/EMito-Metrix>

You must upload these following files from the GitHub repository (!! Keep the same file names !!) :

- EMito_Metrix_Fiji_.ijm (Fiji plugin file)
- dataComputation.py (python file for data display)
- mlComputation.py (python file for data computation and prediction)
- requirements.txt (python package dependencies)
- GeneralistModel_GM_EM.zip (Cellpose GM model)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_FLY_EM.zip (Cellpose SM model for Fly)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_HUMAN_EM.zip (Cellpose SM model for Human)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_MOUSE_EM.zip (Cellpose SM model for Mouse)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_ZEBRAFISH_EM.zip (Cellpose SM model for Zebrafish)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

3- EMITOMETRIX PYTHON ENVIRONMENT: INSTALLATION & SETTING

- **Homebrew package installation**

Before creating the Emitometrix python virtual environment, we need to **install the Homebrew package for MacOS**. In your anaconda prompt, copy and past the following command line and press Enter:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

You will need to set your session password during the application installation. Once the application is installed, you will need to **add homebrew to your PATH**. To achieve this, type in your anaconda prompt the python commands as indicated in the “Next steps” section of your terminal.

Then, in your terminal, copy and paste the following command line to **install the limbo package**:

```
brew install libomp
```

- **Create the Emitometrix Python environment**

We are going to create a **Python virtual environment (venv) for data computation plugins**.

A virtual environment is simply a Python subdirectory with its own file structure, python packages and settings. This Virtual environment let you have a stable and reproducible environment for data visualization and computation plugins, which are essential for the proper running of the EMito-Metrix application.

To create the emitometrix venv, type the following python command line in your anaconda prompt, and press Enter:

```
conda create --name emitometrix python=3.11
```

This command line **will create a new python3.11 environment named *emitometrix***.

This will also create a new folder, containing your emitometrix python environment. To get the path of your emitometrix environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda info --envs
```

That will list the paths of all your environments. This emitometrix folder is mostly found under the following Windows path:

/opt/anaconda3/envs/emitometrix

- **Emitometrix environment activation**

To activate your *emitometrix* python environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda activate emitometrix
```

- **Python dependencies installation**

EMito-Metrix plugin relies on the following Python packages, which are not automatically installed during emitometrix python environment installation:

- cloudpickle==3.0.0	- matplotlib==3.9.0	- pyparsing==3.1.2	- six==1.16.0
- contourpy==1.2.1	- numba==0.60.0	- python-dateutil==2.9.0.post0	- slicer==0.0.8
- cycler==0.12.1	- numpy==1.26.4	- pytz==2024.1	- threadpoolctl==3.5.0
- fonttools==4.53.0	- packaging==24.1	- scikit-learn==1.5.1	- tqdm==4.66.4
- joblib==1.4.2	- pandas==2.2.2	- scipy==1.14.0	- tzdata==2024.1
- kiwisolver==1.4.5	- pillow==10.4.0	- seaborn==0.13.2	- umap-learn==0.5.6
- llvmlite==0.43.0	- pynndescent==0.5.13	- shap==0.46.0	- xgboost==2.1.0

The list of these packages is included in the *requirements.txt* file that is found under your *EMito_Metrix* folder

example of EMitometrix folder : /Application/Fiji.app/plugins/EMito_Metrix

To install python dependencies in your emitometrix environment, type the following python command line from your opened anaconda prompt, and press Enter:

```
pip install -r yourFijiFolder/plugins/EMito_Metrix/requirements.txt
```

If you have some issues with this python command line, type the following command line from your anaconda prompt, and press Enter:

```
pip install cloudpickle==3.0.0 contourpy==1.2.1 cycler==0.12.1 fonttools==4.53.0
joblib==1.4.2 kiwisolver==1.4.5 llvmlite==0.43.0 matplotlib==3.9.0 numba==0.60.0
numpy==1.26.4 packaging==24.1 pandas==2.2.2 pillow==10.4.0 pynndescent==0.5.13
pyparsing==3.1.2 python-dateutil==2.9.0.post0 pytz==2024.1 scikit-learn==1.5.1
scipy==1.14.0 seaborn==0.13.2 shap==0.46.0 six==1.16.0 slicer==0.0.8
threadpoolctl==3.5.0 tqdm==4.66.4 tzdata==2024.1 umap-learn==0.5.6 xgboost==2.1.0
```

- **Cellpose environment deactivation**

Once the Cellpose installation is correct, **deactivate your Emitomatrix python environment** typing in your opened anaconda prompt the following python command line:

```
conda deactivate
```

Once you correctly installed these software and environments, the EMito-Matrix application is ready for running. See the *EMitoMatrix tutorial* for running instructions.

EMITO-METRIX INSTALLATION FOR LINUX

1- CELLOPOSE PYTHON ENVIRONMENT: INSTALLATION AND SETTING

- **Anaconda distribution installation**

Install an Anaconda distribution for Python 3 using the following URL:

<https://www.anaconda.com/download/success>

Download the latest version of Anaconda installer, depending on your Linux architecture.

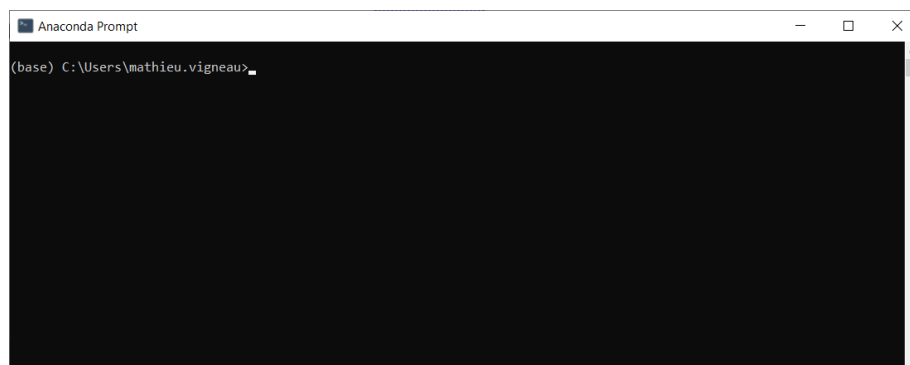
See [here](#) for detailed instructions of Anaconda installation and settings for Linux operating system

- **Cellpose installation & setting**

Once you installed anaconda environment, you will need to create and set a python virtual environment for Cellpose application (which is essential for the EMito-Metrix application).

- **Open an anaconda prompt**

Open a terminal window. This will open a new command prompt window with Anaconda enabled (as shown below).



- **Create the Cellpose Python environment**

We are going to create a **Python virtual environment (venv) for Cellpose** application.

A virtual environment is simply a Python subdirectory with its own file structure, python packages and settings. This Virtual environment let you have a stable and reproducible Cellpose environment, which is essential for the proper running of the EMito-Metrix application.

To create the Cellpose venv, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda create --name cellpose python=3.9
```

This command line **will create a new python3.9 environment named *cellpose***.

This will also create a new folder, containing your cellpose python environment. To get the path of your cellpose environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda info --envs
```

That will list the paths of all you environments. This cellpose folder is mostly found under the following Linux path:

/home/username/anaconda3/envs/cellpose

- **Cellpose environment activation**

To activate your Cellpose python environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda activate cellpose
```

```
Anaconda Prompt - "C:\ProgramData\anaconda3\condabin\conda.bat" activate cellpose
vc          pkgs/main/win-64::vc-14.2-h2eaa2aa_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.29.30133-h43f2093_3
wheel       pkgs/main/win-64::wheel-0.43.0-py311haa95532_0
xz          pkgs/main/win-64::xz-5.4.6-h8cc25b3_1
zlib        pkgs/main/win-64::zlib-1.2.13-h8cc25b3_1

Proceed ([Y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate cellpose
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\mathieu.vigneau>conda create --name cellpose python=3.11
(base) C:\Users\mathieu.vigneau>conda activate cellpose
(cellpose) C:\Users\mathieu.vigneau>
```

- **Cellpose package installation**

Next, you need to install Cellpose application from your Cellpose python venv. In your opened anaconda prompt, type the following python command line, and press Enter:

```
python -m pip install cellpose[gui]
```

This command line will install Cellpose python packages in your python virtual environment, also including the **Cellpose Graphical user Interface**.

- **(Optional) Cellpose GPU package installation**

If you have a **CUDA supported GPU** installed on your computer (see [here](#) for CUDA GPU compatibility), you can install a GPU version of Cellpose.

Please see this [URL](#) for the install instructions of CUDA toolkit and GPU version of Cellpose.

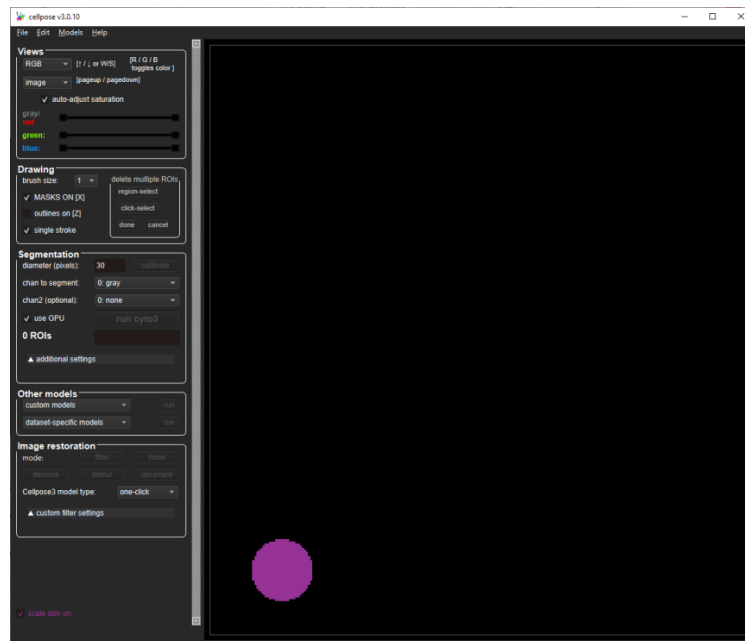
If you need more information about Cellpose detailed requirements, dependencies and installation instructions, see [here](#).

- **(Optional) Cellpose package checking**

Once you installed the Cellpose package, we recommend verifying the validity of the installation. From your Cellpose python environment, type the following python command lines and press Enter:

```
cellpose
```

If the package installation is correct, the following window will appear:



If you have issues with installation, see the cellpose [docs](#) for more details.

- **Cellpose environment deactivation**

Once the Cellpose installation is correct, **deactivate your Cellpose python environment** typing in your opened anaconda prompt the following python command line:

```
conda deactivate
```

2- FIJI INSTALLATION AND SETTING

- Fiji software installation

Install Fiji software distribution using this following URL:

<https://imagej.net/software/fiji/downloads>

Download the latest version of Fiji for Linux 64-bit using the *imagej.net (USA)* link. After downloading, go to your download folder and double-click the *Fiji-zip* file to uncompress it.

See [here](#) for detailed installation instructions and system requirements of Fiji application

- FIJI plugin updating & setting

EMito-Metrix plugin relies on the following Fiji plugins, which are not automatically installed during the Fiji software installation:

- Bio-Formats (<https://imagej.net/formats/bio-formats>)

The Bio-Formats Importer is a plugin for reading data into Fiji. It can open many dozens of proprietary life sciences formats

- ImageScience (<https://imagej.net/libs/imagescience>)

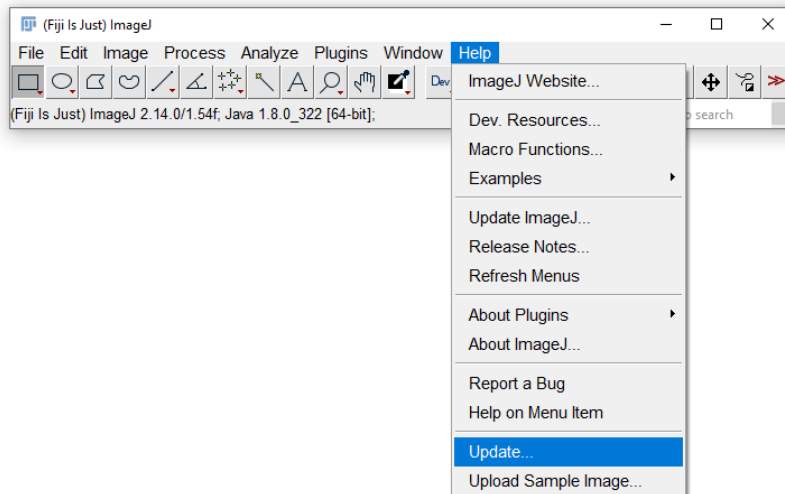
ImageScience is a library of Java classes to facilitate scientific image computing.

- PTBIOP (<https://wiki-biop.epfl.ch/en/ipa/fiji/update-site>)

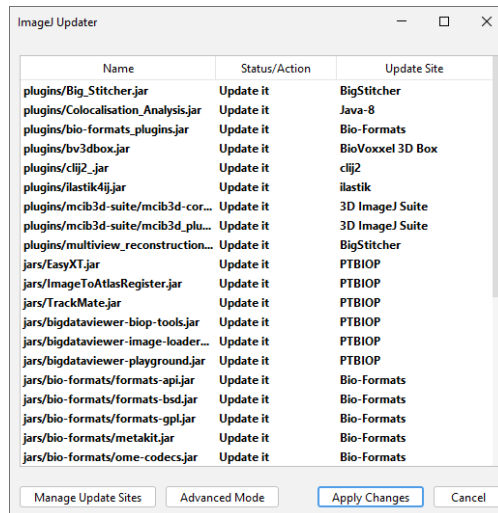
The EPFL - PTBIOP (BioImaging & Optics Platform) is a Fiji package including plugins, macros, ActionBars and Wrappers for external software calls (Cellpose, Elastix, Ilastix...)

- IBMP CNRS

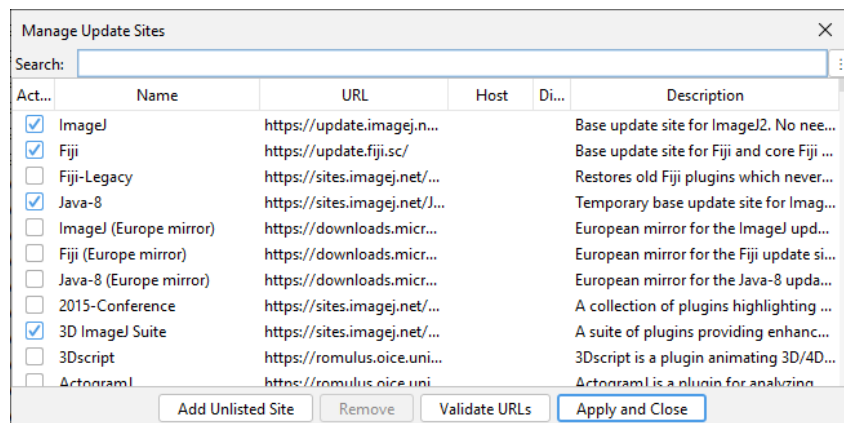
- In your Fiji App directory, double-click the *ImageJ-win64.exe* file to start Fiji software
- Once the software is opened, select *update* from the *Fiji Help menu* (as shown below) in order to update Fiji plugins



- In the *ImageJ Updater* window, select the *Manage update sites* button (see below).



- The *Manage Update Sites* window appears, which contains a list of Fiji plugins you can upload and install in your Fiji application (plugins already installed are checked).



From this *Manage Update Sites* window, check the following plugins to install it:

- Bio-Formats
- IBMP-CNRS
- ImageScience
- PTBIOP

- Once these plugins are checked, click on *Apply and Close* button from the *Manage Update Sites* window. Then, click on *Apply Changes* button from the *ImageJ Updater* window
- Once the plugin are installed, restart Fiji application

- **EMito-Metrix plugin installation & setting**

- **EMito_Metrix folder creation**

In your Fiji plugins folder (*yourFijiFolder\plugins*), create a new folder named **EMito_Metrix**

- **EMito_Metrix file upload**

Download and save the EMito_Metrix files in your *EMito_Metrix Fiji folder*, using the following GitHub repository:

<https://github.com/Restore-Lab/EMito-Metrix>

You must upload these following files from the GitHub repository (!! Keep the same file names !!) :

- EMito_Metrix_Fiji_.ijm (Fiji plugin file)
- dataComputation.py (python file for data display)
- mlComputation.py (python file for data computation and prediction)
- requirements.txt (python package dependencies)
- GeneralistModel_GM_EM.zip (Cellpose GM model)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_FLY_EM.zip (Cellpose SM model for Fly)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_HUMAN_EM.zip (Cellpose SM model for Human)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_MOUSE_EM.zip (Cellpose SM model for Mouse)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

- SpecialistModel_SM_ZEBRAFISH_EM.zip (Cellpose SM model for Zebrafish)

Once copied, unzip the file in the EMito_Metrix folder and delete the zip archive

3- EMITOMETRIX PYTHON ENVIRONMENT: INSTALLATION & SETTING

- **Create the Emitometrix Python environment**

We are going to create a **Python virtual environment (venv)** for data computation plugins.

A virtual environment is simply a Python subdirectory with its own file structure, python packages and settings. This Virtual environment let you have a stable and reproducible environment for data visualization and computation plugins, which are essential for the proper running of the EMito-Metrix application.

To create the emitometrix venv, type the following python command line in your anaconda prompt, and press Enter:

```
conda create --name emitometrix python=3.11
```

This command line **will create a new python3.11 environment named *emitometrix***.

This will also create a new folder, containing your emitometrix python environment. To get the path of your emitometrix environment, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda info --envs
```

That will list the paths of all you environments. This emitometrix folder is mostly found under the following Linux path:

/home/username/anaconda3/envs/emitometrix/

- **Emitometrix environment activation**

To activate your ***emitometrix* python environment**, type the following python command line in your opened anaconda prompt, and press Enter:

```
conda activate emitometrix
```

- **Python dependencies installation**

EMito-Metrix plugin relies on the following Python packages, which are not automatically installed during emitometrix python environment installation:

- cloudpickle==3.0.0	- matplotlib==3.9.0	- pyparsing==3.1.2	- six==1.16.0
- contourpy==1.2.1	- numba==0.60.0	- python-dateutil==2.9.0.post0	- slicer==0.0.8
- cycler==0.12.1	- numpy==1.26.4	- pytz==2024.1	- threadpoolctl==3.5.0
- fonttools==4.53.0	- packaging==24.1	- scikit-learn==1.5.1	- tqdm==4.66.4
- joblib==1.4.2	- pandas==2.2.2	- scipy==1.14.0	- tzdata==2024.1
- kiwisolver==1.4.5	- pillow==10.4.0	- seaborn==0.13.2	- umap-learn==0.5.6
- llvmlite==0.43.0	- pynndescent==0.5.13	- shap==0.46.0	- xgboost==2.1.0

The list of these packages is included in the *requirements.txt* file that is found under your *EMito_Metrix Fiji* folder

example of EMitometrix folder : /home/username/Fiji.app/plugins/EMito_Metrix

To install python dependencies in your emitometrix environment, type the following python command line from your opened anaconda prompt, and press Enter:

```
pip install -r yourFijiFolfer\plugins\EMito_Metrix\requirements.txt
```

- **Cellpose environment deactivation**

Once the Cellpose installation is correct, **deactivate your Emitometrix python environment** typing in your opened anaconda prompt the following python command line:

```
conda deactivate
```

Once you correctly installed these software and environments, the EMito-Metrix application is ready for running. See the *EMitoMetrix tutorial* for running instructions.