Cell-ACDC

A **GUI**-based framework for **segmentation**, **tracking** and **cell cycle annotations** of microscopy imaging data. It includes two of the latest deep learning methods, [Cellpose](https://cellpose.readthedocs.io/en/latest/) and [YeaZ](https://github.com/lpbsscientist/YeaZ-GUI).

*Written in Python 3 by Francesco Padovani and Benedikt Mairhoermann.*

Checkout our paper here.

# Installation

1. Download the latest release from [here](https://github.com/SchmollerLab/Cell_ACDC/releases).
2. If you don’t already have Python or Anaconda, download and install Miniconda for Python 3.8 [here](https://docs.conda.io/en/latest/miniconda.html). We recommend using Anaconda even you already have Python.
3. Follow the instructions below specific to your OS

## Installing on Windows using conda

1. Unzip the latest release you downloaded before. For this example, I will assume it was unzipped into C:\Users\Frank
2. Open the Anaconda Prompt (you should be able to find it from the search bar)
3. Navigate to the folder where you unzipped Cell\_ACDC, (in this example it is C:\Users\Frank\Cell-ACDC ) by typing cd “C:\Users\Frank\Cell\_ACDC” Press “Enter” to confirm. Note that if you unzipped into a drive different from C:\ tou first need to change the drive letter in your terminal. To do so type the letter first (e.g., G:) and then you can navigate with the cd command.

Text

Description automatically generated

1. Now type the following commands **one at the time** (press “Enter” after each command and type “Y” when requested):

conda update -n base -c defaults conda

conda clean --all

conda env create --file acdc.yml

Anaconda will create the environment with Python 3.8 and all the packages required. This step can take several minutes (about 20 minutes if I have to guess, but it depends on your internet connection speed).

If successful, your terminal should now look like the screenshot below (red circle around the part that will tell you that the installation was successful). If you had an error, you could try installing using pip (see instructions below) or open an issue [here](https://github.com/SchmollerLab/Cell_ACDC/issues).

Text

Description automatically generated

## Installing on Windows using pip

1. Download and intall Python 3.8.4 from [here](https://www.python.org/ftp/python/3.8.4/python-3.8.4-amd64.exe). **Make sure to check the option** Add Python 3.8 to PATH and then install with default options.

Graphical user interface, text, application

Description automatically generated

1. Unzip the latest release you downloaded before. For this example, I will assume it was unzipped into C:\Users\Frank
2. Open a terminal (either a Command Prompt or PowerShell, you can find both from the search bar)
3. Navigate to the folder where you unzipped Cell\_ACDC, (in this example it is C:\Users\Frank\Cell\_ACDC) by typing cd “C:\Users\Frank\Cell\_ACDC” Press *Enter* to confirm. Note that if you unzipped into a drive different from C:\ tou first need to change the drive letter in your terminal. To do so type the letter first (e.g., G:) and then you can navigate with the cd command.
4. Now type the following commands **one at the time** (press “Enter” after each command and type “Y” when requested):

py -m pip install --upgrade pip

py -m venv env

.\env\Scripts\activate

py -m pip install -r requirements.txt

You will now see all the required packages being installed. If successful, your terminal should look like the screenshot below (you can ignore that warning). If you had an error, you could try installing using Anaconda (see instructions above) or open an issue [here](https://github.com/SchmollerLab/Cell_ACDC/issues).Text

Description automatically generated

## Installing on macOS using conda

1. Unzip the latest release you downloaded before. For this example, I will assume it was unzipped into SCREENSHOTPATH
2. Open a **Terminal** (Click the Launchpad icon Apps in Launchpad am Mac sortieren - Macwelt in the Dock, type “Terminal” in the search field, then click Terminal)
3. Navigate to the folder where you unzipped Cell\_ACDC, (in this example it is SCREENSHOTPATH/Cell\_ACDC) by typing cd “SCREENSHOTPATH/Cell\_ACDC” Press “Enter” to confirm.

**SCREENSHOT**

1. Now type the following commands **one at the time** (press “Enter” after each command and type “Y” when requested):

conda update -n base -c defaults conda

conda clean --all

conda env create --file acdc.yml

Anaconda will create the environment with Python 3.8 and all the packages required. This step can take several minutes (about 20 minutes if I have to guess, but it depends on your internet connection speed).

If successful, your terminal should now look like the screenshot below (red circle around the part that will tell you that the installation was successful). If you had an error, you could try installing using pip (see instructions below) or open an issue [here](https://github.com/SchmollerLab/Cell_ACDC/issues).

**SCREENSHOT**

## Installing on macOS using pip

1. Download and install Python 3.8.5 from [here](https://www.python.org/ftp/python/3.8.5/python-3.8.5-macosx10.9.pkg). Install with default options.
2. Unzip the latest release you downloaded before. For this example, I will assume it was unzipped into /Users/anikavanessaseel/Documents/GitHub/Cell\_ACDC
3. Open a **Terminal** (Click the Launchpad icon Apps in Launchpad am Mac sortieren - Macwelt in the Dock, type “Terminal” in the search field, then click Terminal)
4. Navigate to the folder where you unzipped Cell\_ACDC, (in this example it is /Users/anikavanessaseel/Documents/GitHub) by typing cd “/Users/anikavanessaseel/Documents/GitHub/Cell\_ACDC” Press *Enter* to confirm.
5. Now type the following commands **one at the time** (press “Enter” after each command and type “Y” when requested):

python3 -m pip install --user --upgrade pip

python3 -m venv env

source env/bin/activate

python3 -m pip install -r requirements.txt

You will now see all the required packages being installed. If successful, your terminal should look like the screenshot below. If you had an error, you could try installing using Anaconda (see instructions above) or open an issue [here](https://github.com/SchmollerLab/Cell_ACDC/issues).

Text

Description automatically generated

## Installing on Linux using conda

1. Open a **Terminal**
2. Unzip the latest release you downloaded before. For this example, I will assume it was unzipped into /home/elpado/GitHub/
3. Navigate to the folder where you unzipped Cell\_ACDC, (in this example it is /home/elpado/GitHub/) by typing cd “/home/elpado/GitHub/Cell\_ACDC”
4. Now type the following commands **one at the time** (press “Enter” after each command and type “Y” when requested):

conda update -n base -c defaults conda

conda clean --all

conda env create --file acdc.yml

Anaconda will create the environment with Python 3.8 and all the packages required. This step can take several minutes (about 20 minutes if I have to guess, but it depends on your internet connection speed).

If successful, your terminal should now look like the screenshot below (red circle around the part that will tell you that the installation was successful). If you had an error, you could try installing using pip (see instructions below) or open an issue [here](https://github.com/SchmollerLab/Cell_ACDC/issues).

## Installing on Linux using pip

1. Open a Terminal
2. Make sure you have Python 3.8 and pip installed. Check if you have Python with python –version command and check if you have pip with pip help command. If you don’t have them install with the following commands:

sudo apt-get update

sudo apt-get install python3.8

sudo apt-get install python3-pip

1. Unzip the latest release you downloaded before. For this example, I will assume it was unzipped into /home/elpado/GitHub/
2. Navigate to the folder where you unzipped Cell\_ACDC, (in this example it is /home/elpado/GitHub/) by typing cd “/home/elpado/GitHub/Cell\_ACDC”
3. Now type the following commands **one at the time** (press “Enter” after each command and type “Y” when requested):

python3 -m pip install --user --upgrade pip

python3 -m venv env

source env/bin/activate

python3 -m pip install -r requirements.txt

You will now see all the required packages being installed. If you had an error, you could try installing using Anaconda (see instructions above) or open an issue [here](https://github.com/SchmollerLab/Cell_ACDC/issues).

# First steps

## Starting the main launcher

1. Open a terminal:
   * Windows: **Anaconda Prompt** if you installed with conda
   * Windows: Command Prompt or **PowerShell** if you installed with pip
   * Unix/maxOS: **Terminal**
2. Navigate to the Cell-ACDC folder with the command cd like you did when you installed it.
3. **Activate** the environment:
   * Conda: conda activate acdc
   * pip on Windows: .\env\Scripts\activate
   * pip on Unix/macOS: source env/bin/activate
4. Run the main launcher:
   * Windows: python main.py
   * Unix/macOS: python3 main.py
5. If you get the error ImportError: No module named 'Tkinter' you need to install tkinter with the command sudo apt-get install python3-tk
6. The first time, it will take 1 or 2 minutes to launch. The next times it will be faster. Once launched, you should get the following Welcome Guide window.

Graphical user interface, text, application, email

Description automatically generated

## Load microscopy file

To load a microscopy file, Cell-ACDC uses the java library [Bio-Formats](https://www.openmicroscopy.org/bio-formats/) and the Python library [python-bioformats](https://pythonhosted.org/python-bioformats/). The python-bioformats library was developed for [CellProfiler](https://cellprofiler.org/) and it is **embedded** into Cell-ACDC. It essentially allows you to run the java code from Bio-Formats from Python. Have a look [here](https://docs.openmicroscopy.org/bio-formats/6.7.0/supported-formats.html) for a list of supported file formats.

To load a microscopy file into the Cell-ACDC pipeline we first have to **convert it into a specific data structure**. We included a module that allows you to **automatically create the required data structure**. However, if it fails, you can create it manually with ImageJ/Fiji. Read the section of this manual called “[Manually create data structure from microscopy file(s)](#_Manually_create_data)”.

1. From the [main launcher](#_Starting_the_main) (could be behind Welcome Guide window) click on “Create data structure from microscopy file(s)…” button and follow the instructions of the Wizard.

Graphical user interface, text, application, email

Description automatically generated

1. Once the creation of the data structure is finished you are ready to start using your new labelling tool :D. The easiest way to start is from the [Quick Start](#_Quick_Start) section of this User Manual

## Manually create data structure from microscopy file(s)

1. If you don’t have it, download and install Fiji from [here](https://imagej.net/software/fiji/)
2. Open Fiji app and launch the Bio-Formats importer from the menu “Plugins 🡪 Bio-Formats 🡪 Bio-Formats Importer” and select your microscopy file (one at the time).
3. Check the options “Use virtual stack” and “Split channels” as in the screenshot below

Graphical user interface, text, application, email

Description automatically generated

1. If you have multiple positions (series) in the file you opened, you will be asked to select a position. We recommend opening one position at the time, to avoid memory issues.
2. You should now have one image window for each channel you had in the file. Select the window with the image data from the first channel (the window name should be something like “filename … C=0”), then “File 🡪 Save as… 🡪 Tiff”
3. As a filename we recommend calling it with the same name of the original microscopy file (if it’s not too long) **WITHOUT the extension** plus something like “\_channel0.tif” where instead of “channel0” you can write whatever you like (e.g., DAPI or GFP etc.). So for example a .czi (Zeiss microscope) file called ASY15-1\_15nM-01.czi can be save as ASY15-1\_15nM-01\_phase\_contr.tif
4. Save the .tif file to a path called “/Position\_1/Images”.
5. Repeat 2-7 for all the other positions.
6. In the end you should have the following folder structure:

Experiment\_folder

Position\_1

Images

basename1\_channel0.tif

basename1\_channel1.tif

…

basename1\_channeln.tif

Position\_2

Images

basename2\_channel0.tif

basename2\_channel1.tif

…

basename2\_channeln.tif

…

Position\_n

Images

basenamen\_channel0.tif

basenamen\_channel1.tif

…

basename2\_channeln.tif

1. Once the creation of the data structure is finished you are ready to start using your new labelling tool :D. The easiest way to start is from the [Quick Start](#_Quick_Start) section of this User Manual

# Quick Start

Cell-ACDC is composed of **three main modules**:

* **Data prep**: align time-lapse data, crop, and select a z-slice or z-projection and a ROI for segmentation. [More details…](#_Data_Prep_module)
* **Segmentation**: to automatically segment multiple experiments and multiple positions with the embedded deep learning models ([YeaZ](https://www.nature.com/articles/s41467-020-19557-4) for yeast cells and [Cellpose](https://cellpose.readthedocs.io/en/latest/installation.html) for multiple model organisms). [More details…](#_Segmentation_module)
* **Main GUI**: to visualize segmentation masks, correct segmentation and tracking errors, and cell cycle annotations. [More details…](#_Main_GUI)

The easiest way to start is to **open the main GUI**. Next, if you already created the data structure (see [Load microscopy file](#_Load_microscopy_file) section) you can click on the “Open Folder” button on the toolbar, otherwise you can go to “File 🡪Open image/video file…”. To start the main GUI, click the “Launch GUI…” button on the main launcher.

Graphical user interface, text, application, email

Description automatically generated

# Data Prep module

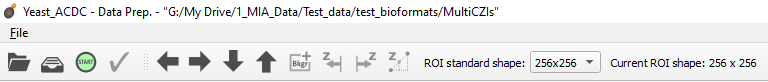
To use the data prep module, you need to **first create the required data structure.** See [this](#_Load_microscopy_file) section.

Use the **Data Prep** module if you need to do one of the following **tasks**:

1. Select a **z-slice** or **z-projection** for segmentation of 3D z-stacks
2. **Align frames** of time-lapse microscopy data (RECOMMENDED, it is revertible)
3. Calculate **background metrics** (median, mean etc.) from one or more **rectangular areas**. The median will be used later for background subtraction. The areas are movable and resizable.
4. Select a region of interest (**ROI**) for segmentation
5. **Crop** images to reduce memory usage (RECOMMENDED, if possible)

## Loading data

1. **Launch** the data prep module, click on the “1. Launch data prep module…” button on the main launcher.
2. Click on the “**Open Folder**” button on the toolbar.



1. Select a specific Position folder or the entire experiment folder.
2. Follow the instructions in the pop-up windows. Make sure to enter the **correct metadata.**

*NOTE: For time-lapse microscopy you can load only one position at the time. Select multiple positions only if you have single 3D z-stacks or 2D images.*

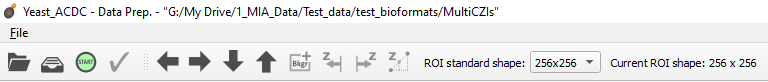
## Usage

1. If your data does not contain 3D z-stacks go to point 2. Otherwise, you can visualize the **z-slices** with the **scrollbar** below the image or choose a **z-projection** method with the selector on the right side of the scrollbar.

Every time you change visualization method, the system will save it. It will then assume that the **last visualization is the preferred one** and it will be used for **segmentation**.

For **time-lapse data** you have **additional buttons** to help the selection. Go to the section “Additional functions” for details about their functionality.

1. If you do not need to select a ROI, crop, align or calculate background metrics you can close the window. Otherwise press the “**Start**” button on the toolbar and follow the instructions on the pop-up windows.



1. The GUI now will be **unresponsive** until the process terminates, so do not close it. You can follow **progress** in the **terminal**. Once it finishes a **red rectangle** will appear, along with a grey rectangle (see screenshot below). If you do not need to select a **ROI,** calculate **background metrics,** or crop you can close the window now, otherwise go to the next point.

A screenshot of a computer

Description automatically generated with low confidence

1. The **red ROI** is used for either **cropping** or saving the coordinates where to compute **segmentation**. The **grey ROI** (Bkgr. ROI) is used to calculate **background metrics** from that area (median, mean, quantiles etc.). You can add more background ROIs with the ‘Add ROI where to calculate background intensity’ button on the toolbar.
2. **Resize** and **move** the ROIs until you are happy with their position and size, click on the **green tick** button on the toolbar, then follow instructions in the pop-up windows. The GUI will be **unresponsive** until the process terminates, so **do not close it**. You can check progress in the terminal.

## Additional functions

|  |  |  |
| --- | --- | --- |
|  | Go to **previous/next** position or frame (time-point). | |
|  | Go **10** positions or frames **backward/forward.** | |
|  | Use the same **z-slice** from current frame to all past/future frames. | |
|  | Use **linearly interpolated z-slices** from first frame to current frame. | |
|  | | Select one of the **standard shapes** for the red ROI |

# Segmentation module

The segmentation module is used for **automatically segmenting multiple experiments and multiple positions** in one session.

To use the segmentation module, you need to **first create the required data structure.** See [this](#_Load_microscopy_file) section.

*NOTE: if you are just testing you can also segment in the main GUI. Use this module when you need to segment many experiments and/or many positions.*

## Usage

This module is very easy to use, you simply have to follow the **instructions** in the **pop-up windows**. To **launch** the module, click on “2. Launch segmentation module…” button on the main launcher.

Graphical user interface, text, application, email

Description automatically generated

# Main GUI