# Overview

This software allows manual correction of segmentation labels in 2D and 3D images. It was developed specifically for correcting segmentation labels of chromosomes in cell nuclei but can be applied to other areas outside Molecular Biology. Manual correction is a necessary post-processing step of many automatic segmentation algorithms, which fail to segment objects perfectly, and require merging, splitting, and manual delineating. The software can also be used for generating ground truth segmentation masks for training supervised machine learning models.

# Installation

## System requirements

* Windows: tested on Python 2.7, 3.6, and 3.7 but may work on other versions
* Linux: tested on Python 3.4 - 3.11 version but may work on earlier ones.
* Mac OS: tested on Python 3.10 but should work on any Python 3 version.

To download python libraries, we recommend using PIP.

Specific installation for different operating systems is outlined below. Please ensure that no errors are reported at the installation step. If there are errors you are unable to fix, please let us know.

## Windows

Pyfix3d was developed on Windows and extensively tested, so we recommend using it on Windows if possible.

1. Make sure you have Visual C++ Development Tools version 14.0 or higher.
2. Install the required external libraries in a terminal:

**python -m pip install tifffile pysimplegui pynput==1.6.0 pptk --find-links .**

Note that “**python**” may have to be replaced **by “python3”, “py”, “python36”**, etc. depending on the alias used by the installation.

1. Then start the application from the terminal:

**python path/to/project/Pyfix3d/pyfix3d.py**

specifying the correct path. Alternatively, first navigate to the Pyfix3d project and open the terminal from that location, then type:

**python pyfix3d.py**

## Linux

The program has been tested on Ubuntu 17.10 and Ubuntu 22.04.

1. Navigate to the project directory and install the required external libraries in a terminal:

**python -m pip install tifffile pysimplegui pynput==1.6.0 pptk --find-links .**

1. Follow the steps below to link a dependency for the PPTK library (replace the location in the first line with the correct location of your python site-packages)

**cd /home/ubuntu/.pyenv/versions/3.6.8/lib/python3.6/site-packages/pptk/libs/**

**mv libz.so.1 libz.so.1.old**

**sudo ln -s /lib/x86\_64-linux-gnu/libz.so.1**

**export QT\_DEBUG\_PLUGINS=1**

1. Install the required Xinerama library

**sudo apt-get libxcb-xinerama0**

1. Then start the application:

**python pyfix3d.py**

## Mac

The program has been tested on Mac OS Catalina with python 3.10.

1. Navigate to the project directory and install the required external libraries in a terminal:

**python -m pip install tifffile pysimplegui pynput pptk --find-links .**

1. Then start the application:

**python pyfix3d.py**

# Example workflow

Below is a description of a set of steps demonstrating Pyfix3d functionality. To follow these steps, please use the provided example database.

## Database description

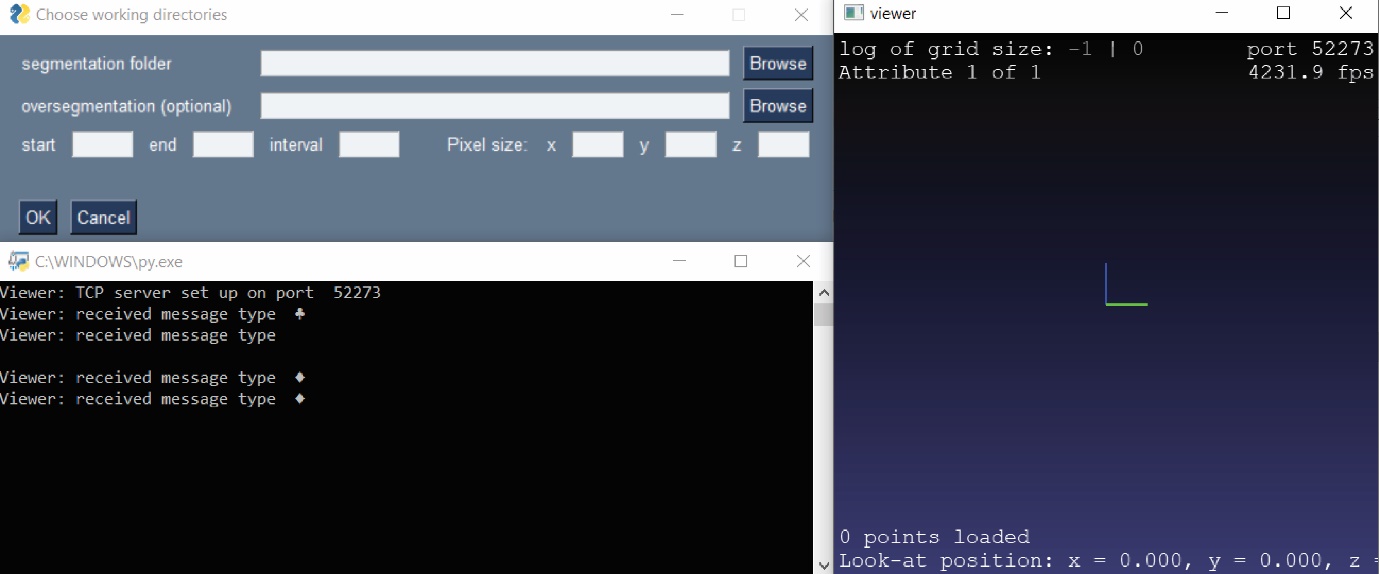
The folder sample\_data contains 13 .tif files. Each .tif file is a 3D stack of a cell nucleus of size (120, 724, 724), i.e. 120 Z planes of size 724x724 pixels. These are different time points in a time lapse experiment, 8 seconds apart. The images represent chromosomes which are segmented and tracked over time. Each chromosome is a 3D object denoted with an integer. For example, all pixels of value 31 constitute one chromosome, and all pixels of value 40 constitute another. The visualizer will assign different colors to different numbers, and the chromosomes will appear as differently colored objects.

## Usage

To start the program, after installing the required dependencies as explained above, navigate to Pyfix3d folder, start a terminal, and type:

**python pyfix3d.py**

The following three windows will appear:



The window on the top left is the initialization menu. The window on the right is the viewer, which is empty for now. The black window on the bottom left is a terminal, which runs backend operations. We recommend minimizing it and making sure it stays open.

Inside the menu (top left), “Segmentation folder” is the only required argument. Using the Browse button, please navigate to the folder sample\_data. Leaving the other fields blank will load all images from the selected folder, and will assign 1 (arbitrary unit) to the pixel size in x, y, and z. These pixel sizes should be set in case of anisotropic data – where the physical distance between Z planes is different from the pixel size in XY. This is the case with the current data.

Start and end will enable loading a subset of the available images, at specific intervals.

Let’s use the following settings:

start 1100 end 1105 interval 1 Pixel size: x 0.045 y 0.045 z 0.18

These are the pixel sizes in microns of the actual experiment.

After pressing OK, the images will start loading. Upon completion, the following outlook should appear in the viewer:



The sliding bar on the bottom changes the time point.

The following operations are inherited from the PPTK library:

Left click + drag performs rotation.

Scroll performs zooming in/out.

Double left clicking on an object centers the view on that object.

Holding Shift + left click translates the view left/right or up/down.

To select specific labels and hide the rest, we can use the “Find object” button. Pressing “Find object” invokes a pop-up window (left image below), where we can type the number of the labels of interest. Let’s type in the following sequence, separated by comas: 31,41,80,100,101. Clicking OK grays out all other labels, as shown on the right:

A screenshot of a computer

Description automatically generated

There are three types of manual corrections that Pyfix3d supports: merging, splitting, and manual re-coloring.

A computer screen with a computer keyboard

Description automatically generated

### Merging

The light and dark green labels on the first image above are part of the same chromosome. To merge them, let’s do the following sequence of actions:

1. Hold down Ctrl and click on the light label. It will become “active”, appearing yellow (second image above).
2. Hold down Ctrl and click on the dark green label. It will also become active and yellow (third image above).
3. Press “Correction”

The two labels will now be merged, and they will be the same object from now on, adopting the color of the second (fourth image above).

This procedure can be used to merge more than two labels: select all labels to be merged using Ctrl + left click and then click “Correction”. They will be assigned the color of the last selected label.

In case we want to correct all such instances (e.g. merge all blue to the green) over time, we can use “Correct all” instead of “Correct”.

Sometimes, it is desirable to recolor only part of a label, not the whole label. This can be done in the following way:

1. Ctrl + left click on the **target** label (whose color we should use) and press the button “Set/clear correction color”. You should see the color of the button change accordingly.
2. Ctrl + left click on the **source** label (part of which we want to change) and press the button “Set/clear source color”. You should see the color of the button change accordingly (see picture below).
3. Left click with the mouse and drag. You should see the pixels you pass will turn yellow (top right on the picture below). However, since the source color is set, the program will ignore the other labels, and will retain only the pixels belonging to this color. So, when you finish the delineation with the mouse, you will see something like the bottom left in the picture below.
4. Now pressing “Correction” will recolor only this chunk (bottom right).
5. It is important to click the buttons “Set/clear source color” and “Set clear correction color” once again, to clear them.

A screenshot of a computer

Description automatically generated

### Splitting

Splitting a label is equivalent to creating a new label from a subset of its pixels. To do this, we set the **source** color exactly as above (using “Set/clear source color”), then delineating the part of the label we want to recolor and pressing “Make new”. The chunk will be recolored in a new, unused color (i.e. setting it to a unique integer), thus splitting the label.

### Saving

Use the “Save all” button to save the changes. We recommend doing this often, as a safeguard against sudden crash of your machine or the program itself. It may be wise to save the results to a new folder, otherwise the existing images will be overwritten. Note that only the images that were changed will be saved.

### Creating a snapshot

The “Snapshot” button will create a picture of the current view and save it as “snap.png” in the project folder. You can then rename it and transfer it to another location.

### Marking and unmarking objects

Selecting an object with Ctrl + click and pressing the button “Mark” will add this label to the list of marked objects. It will be automatically hidden from the current view (but not deleted). We developed this feature to hide labels which are already corrected, so they get out of the way and allow better visualization of the rest.

The buttons “Show marked” and “Show unmarked” will show show one of the groups, respectively, and hide the rest. “Show hidden” will show all labels.

Note that the list of marked objects is saved as the file “hidden\_objects.txt” inside the project folder. Therefore, if you want to keep the list for later re-use, please copy this file to another location and then transfer it back after re-opening the program.

### Reporting errors

Watch out for the file “error\_log.txt” in the project folder. If it gets created, some exception has happened and has been caught. Please report this to us. If the exception is not caught and the program crashes, there should be some information in the terminal instead. Please report this to help us fix the issues.