

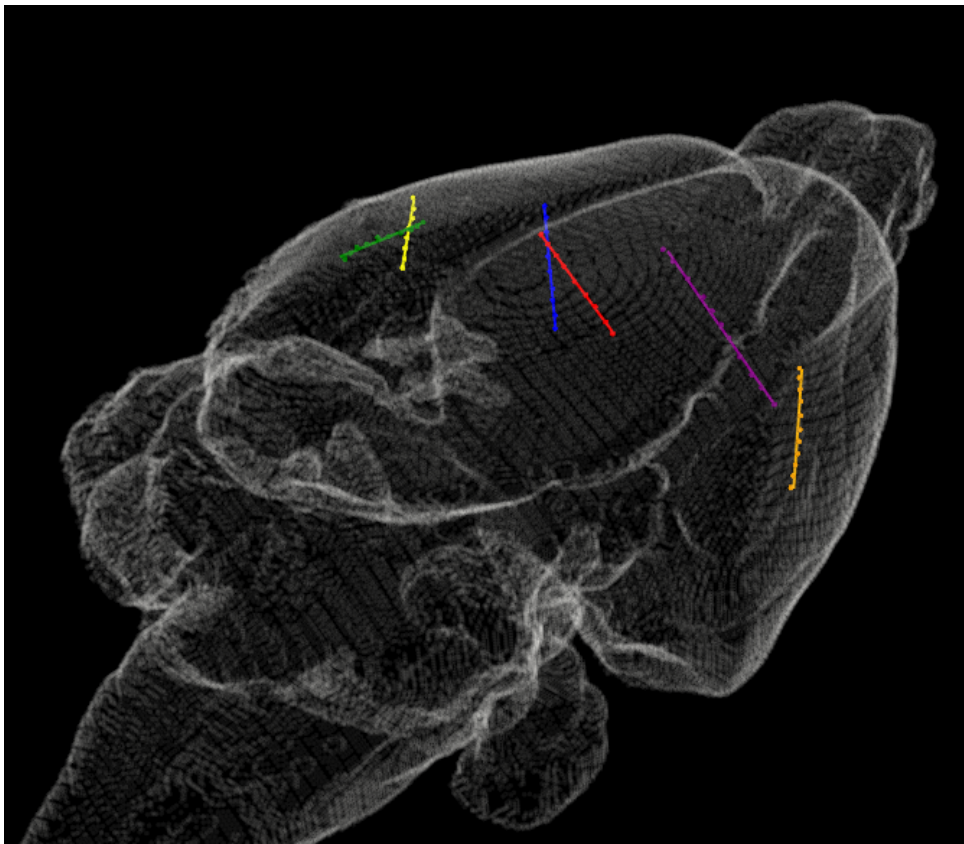
User manual

TRACER-Toolkit for Reconstructing Anatomical Coordinates in Rats

v. 2021 -4

Pearl Saldanha
pearl.saldanha@ntnu.no

August 25, 2021



Introduction

TRACER-3D is a useful toolbox for visualizing the trajectories of recording electrodes (e.g. Neuropixels) through different sub-regions of the rat brain. Anatomical delineations are referenced from the Waxholm Space atlas of the adult Sprague Dawley rat brain (<https://www.nitrc.org/projects/whs-sd-atlas>). There are three packages in the TRACER-3D toolbox. One is for locating electrode tracks in the brain post-hoc using histological images, one is for generating coordinates prior to surgery in order to target specific brain regions, and one for visualizing the virus expression. NOTE: For Neuropixels, we included the 175um tip + 9600um electrode surface length (20um * 960 electrodes * 1/2 row/electrodes). So happy tracing!

Contents

1	Requirements	3
2	Setup of working environments	3
2.1	Anaconda environments	3
2.2	Other environments	4
2.3	Installation of TRACER	4
3	TRACER API	4
3.1	Load Atlas	4
3.2	Generating pre-surgical coordinates	4
3.2.1	Planning probe insertion	5
3.2.2	Visualizing probes in the 3D atlas for insertion	5
3.3	Analysis and retracing of probe trajectories	5
3.3.1	Preprocessing histological Images	5
3.3.2	Registering the probe track and navigating the atlas	6
3.3.3	Visualizing probes in the 3D atlas	8
3.4	Virus expression visualization	8
3.4.1	Registering the virus and navigating the atlas	9
3.4.2	Visualizing virus in the 3D atlas	9

1 Requirements

In order to start working with TRACER you will need:

- TRACER package downloaded on your local computer from <https://github.com/Whitlock-Group/TRACER>
- Individually sliced images of rat brain slices (coronal, sagittal, or horizontal) in .jpg format.
- Python 3.8.x or above (Terminal or Python IDE, e.g. Spyder, Pycharm, VS).
- A computer mouse with scroll wheel and a functional keyboard.
- The atlas files can be downloaded from the NITRC website (<https://www.nitrc.org/projects/whs-sd-atlas>). The files that you should have are:
 - WHS_SD_rat_T2star_v1.01.nii.gz
 - WHS_SD_rat_brainmask_v1.01.nii.gz
 - WHS_SD_rat_atlas_v3.nii.gz
 - WHS_SD_rat_atlas_v3.label

2 Setup of working environments

It is recommended to use Anaconda for non experts in Python, this will help with the downloading of the necessary packages and for the updating of the existing ones. However users can also use PyCharm or any other Python IDEs. TRACER requires users to install Numpy, Scipy, matplotlib, vedo, opencv-python, Pillow, scikit-image, scikit-spatial, nibabel, nilearn, tabulate, keyboard and mplcursors packages as dependencies. The required version specifications of these packages are listed in the file 'requirements.txt' in TRACER.

2.1 Anaconda environments

Anaconda can be downloaded for free from their website (<https://www.anaconda.com>). Once installed Anaconda will already include Spyder. To create a new environment in anaconda:

- go to Environments on the left panel,
- click create in the bottom left,
- name your environment and in Packages select both Python (3.8) and R.

After a new environment is created, the necessary packages need to be installed in order to use TRACER, this is done with the following steps:

- open the terminal by clicking the arrow near the name of you environment,
- navigate to the folder where requirements.txt is
- run `pip install -r requirements.txt`

2.2 Other environments

In PyCharm, users can install the dependence packages accordingly through project interpreter. To access project structure, open Settings/Preferences dialog by pressing Ctrl+Alt+S or by choosing File | Settings for Windows and Linux or PyCharm | Preferences for macOS, then expand the Project node, and select Project Interpreter. The instruction in detail can be found <https://www.jetbrains.com/help/pycharm/configuring-python-interpreter.html>.

In terminal, users can go into the directory that TRACER located and run the command `pip install -r requirements.txt` directly to install the dependencies.

2.3 Installation of TRACER

TRACER can be installed from the GitHub page using the terminal. Select the path you want the package to be stored.

```
1 $gitclone <repository> <path>
2
3 git clone https://github.com/Whitlock-Group/TRACER.git ./
   Workspace
```

or by downloading the entire package as a .zip file on your local computer.

3 TRACER API

3.1 Load Atlas

Once in the console after TRACER has been installed on your local computer and the right path is defined on your console where TRACER is stored. Make sure to check the path of your package, you can do this by running this on your console:

```
1 import os
2 os.chdir("/Users/admin")
```

Once your path of the directory and the folder of your local computer, use the following command to load TRACER.

```
1 from TRACER import *
2 atlas = AtlasLoader('folder_path')
```

where 'folder_path' is the path of the folder where users can save atlas files.

3.2 Generating pre-surgical coordinates

One of the features of TRACER is to help the user to plan the electrode insertion before surgery, showing the inclination and insertion distance necessary to target specific brain's regions. For the insertion planning, create a 'probes' folder where you can save the inserted probe and run the following commands:

```
1 from TRACER import *
2 atlas = AtlasLoader('folder_path', atlas_version = 'x')
3 presurgery = ProbesInsertion(atlas,
4 probe_folder = 'probes_folder_path')
```

NOTE: Refer to `example_code.py` for examples on how to use class functions.

3.2.1 Planning probe insertion

The class file **ProbesInsertion** is essentially useful to plan electrode insertion before any surgeries. Once you run the script you will be displayed with the Waxholm atlas. You can then scroll through the slices until you reach the slice of your satisfaction. Then use the commands displayed on the console to view boundaries and determine where in the brain region you want to insert the electrode. Remember you can add upto 6 probes.

3.2.2 Visualizing probes in the 3D atlas for insertion

Now you have saved and registered points of the probe track(s). You can further visualize your probe tracks in 3D by running the line in the console

```
1 vis3d_presurgery = vis_inserted_probes(atlas, probe_folder)
2 vis3d_presurgery.vis2d()
3 vis3d_presurgery.vis3d()
```

One can also reload saved probes by running the line

```
1 presurgery.reload_probes('Probe1')
```

This script gives you the information about the probe insertion angle, regions transverses by the electrode, the location and the number of channels that can be used for recording neural signals. Close and open a new console every time you run this script.

3.3 Analysis and retracing of probe trajectories

One of TRACER's main task is to help the user with the registration and study of the brain regions traversed by the probe(s).

3.3.1 Preprocessing histological Images

This functionality helps you obtain processed, clear, nice looking histology images with desired sizes and contrast. You should run this file to properly register the histology images. To do so - use the following command

```
1 preprocess_hist = preprocess_histology('path')
```

Make sure your brain slices are all saved in the same folder. It is recommended to have the images with consecutive names (e.g. histology01, histology02, etc). Run `%matplotlib inline` line of code in your console before running the entire script.

NOTE: If you are a Mac user, make sure that Python has the security authorization to access input from keyboard. If you get the error in Figure 1 use the `sudo` command in your terminal to open your desired IDE (Spyder is recommended as it was used for developing the software). Use the command `sudo spyder`. Follow the steps below to properly run the script.

```
Exception in thread Thread-12:
Traceback (most recent call last):
  File "/opt/anaconda3/lib/python3.8/threading.py", line 932, in _bootstrap_inner
    self.run()
  File "/opt/anaconda3/lib/python3.8/threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "/opt/anaconda3/lib/python3.8/site-packages/keyboard/__init__.py", line 294, in listen
    _os_keyboard.listen(self._direct_callback)
  File "/opt/anaconda3/lib/python3.8/site-packages/keyboard/_darwinkeyboard.py", line 430, in listen
    raise OSError("Error 13 - Must be run as administrator")
OSError: Error 13 - Must be run as administrator
```

Figure 1: Error for Mac users when running `Preprocess_Histology.py`.

- Add the correct path for your folder:
`histology_folder = Path('/Users/...../Histology/')`
- Choose the plane of view with the keyboard letters (c for coronal, s for sagittal, h for horizontal).
- Once you have adjusted the histology image using the controls exhibited in the console press 's' on the keyboard to finalize your editing and then save.
- You will find the processed histology image in the folder "Processed", that will be created as sub-folder in the Histology folder.

3.3.2 Registering the probe track and navigating the atlas

This file runs based on the images saved from the pre-processing step. You can then select the corresponding brain slice with the Waxholm atlas, overlay the histology to the atlas, and register the probe. Make sure you have the right path to the preprocessed images obtained from the previous script, and to the other relevant folders.

- Close the console where Preprocess_Histology.py was running and open a new one.

```
1 register_probes = ProbesRegistration(atlas ,
    processed_histology_folder='path')
2 probe_folder = 'path_probes'
```

- Add the path here:
`processed_histology_folder = Path('/Users/....../Histology/processed').`
- There are two figures popping up in the screen and to perfectly overlay your histology image with the atlas you need to click 'r' to activate the overlay points.
- Choose at-least 4 points in the same order for both the histology slice and the atlas image that is on your computer screen. Accuracy in selecting the points at this stage will determine a good fit of the histology image to the atlas. The numbering will help you remember the order in which the points on the atlas and histology slice are clicked.

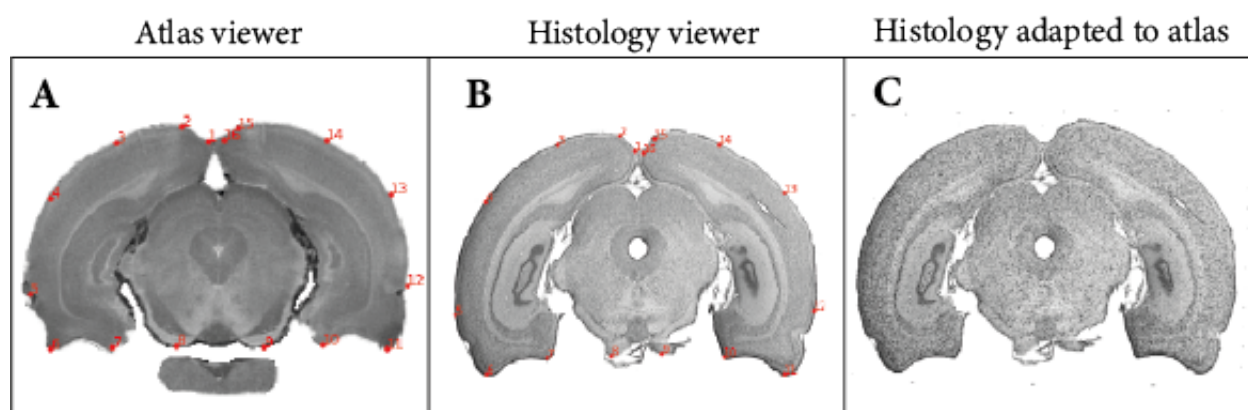


Figure 2: A) shows the numbered clicked points on the atlas. B) shows the Nissel stained version of a histology slice

- Press 'h' to match the histology with the atlas.
- Press 'o' to visualize the regions boundaries of the Waxholm atlas (Figure 3A).

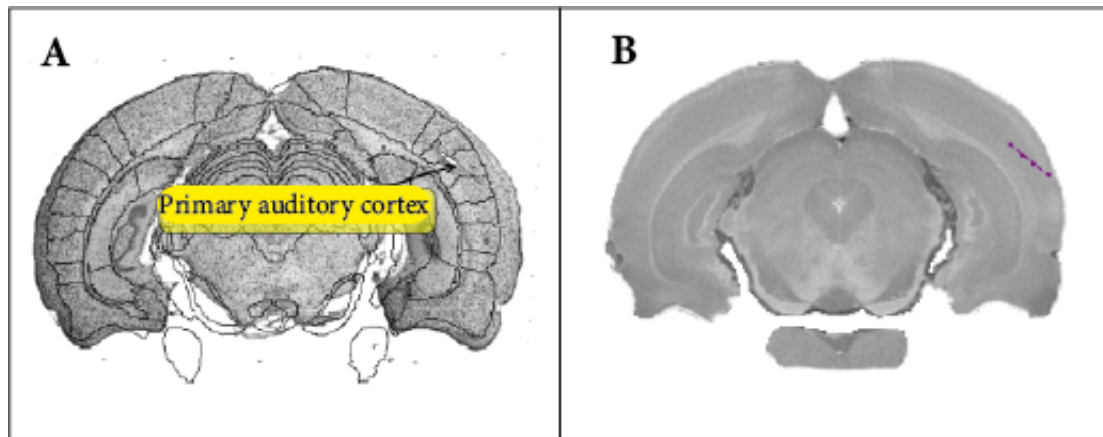


Figure 3: A) Activated grid lines showing the regions of the rat brain in the coronal section. B) Estimated insertion of the probe in the selected rat brain

- After having pressed 'h' and 'o' it is possible to either visualize the regions colors by pressing 'v' or start registering the probes by pressing 'r'.
- Click the points on the probe tracks.
- When registering probes, it is important that the first and the last point clicked correspond to the entry point and to the last visible point of the probe. This applies for all the histology images you want to click the probe points.
- Press 'c' to delete the probe points.
- Press 'w' to visualize probes in '2D' (Figure 3B).
- Press 'e' to save the histology and proceed with the registration of the next slice. The slices will appear according with the ordering given by their names.
- If you have multiple images, once you finish registering the electrode track of one histology image, you'll find the next one lined up on your desktop screen. Follow the same procedure as you did for the first histology image.
- For more options follow the instructions on the console.

NOTE: In case you want to change the background colour of the image or the atlas, you can do it by changing the following: Go to line 21 (Figure 4) in the `Tracker.py` file and change the 'cmap' colour to either white or black depending on your choice of image.

```
21 self.im = ax.imshow(self.X[:, self.ind, :].T, origin="lower", extent=[0, 512*pixdim, 0, 512*pixdim], cmap='gist_yarg')
22 #cmap='gist_yarg' if white background wanted
23 #cmap='gray' for a gray or black background
```

Figure 4: Code to change background colours.

To change the colour of the gridlines of the atlas, go to code line 367 in the `Register_probes.py` file and change to 0 for black, 1 for white or 0.5 for gray.

NOTE: when interacting with keyboard make sure that one figure between Atlas viewer, Histology viewer or Histology adapted to atlas is selected otherwise the the software will not interact with the keyboard.

3.3.3 Visualizing probes in the 3D atlas

Now you have saved and registered points of the probe track(s). You can further visualize your probe tracks in 3D by running:

```
1 vis3dres = vis_registered_probes(atlas,probe_folder)
2 vis3dres.vis3d()
3 vis3dres.vis2d()
```

This script gives you the information about the probe insertion angle, regions transverses by the probes, the location and the number of channels in the probe (e.g. neuropixel) that has been recording.

Analyze purple probe:		
Probe length: 2.18 mm		
Estimated purple probe insertion angle: 8.27 degrees in the anterior direction 38.57 degrees in the lateral direction		
Regions traversed	Initials	Channels
secondary auditory cortex, dorsal area	s a c d a	14
primary auditory cortex	p a c	154
secondary auditory cortex, ventral area	s a c v a	26

Figure 5: Number of recording channels in the electrode.

In the processed folder on your PATH you will then find a .txt file in which you will find relevant information about the probe you have currently visualized (Figure 5).

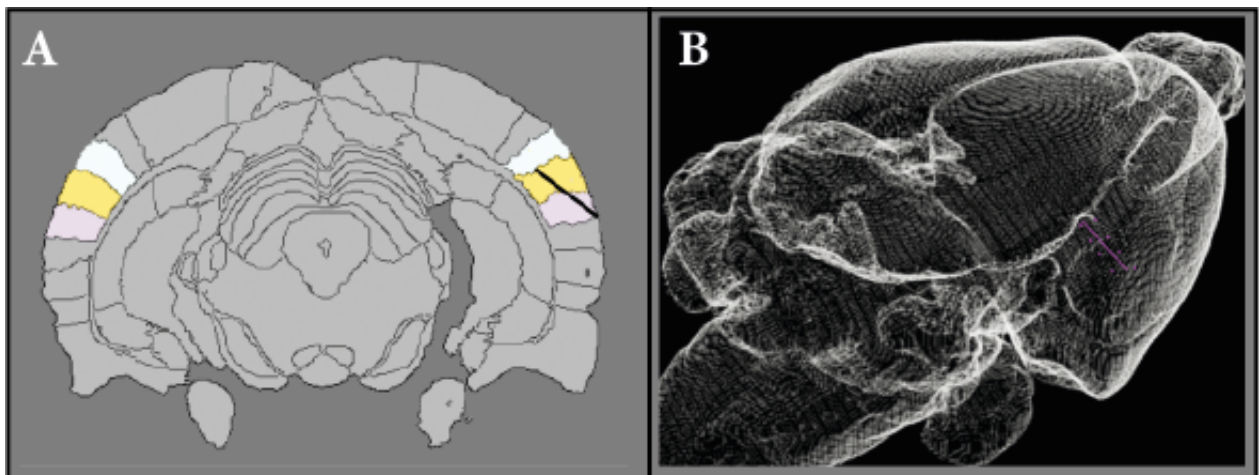


Figure 6: Left: Regions of the rat brain that the electrode has traversed. Right: 3D constructed electrode passing through different regions of the rat brain.

Close and open a new console every time you run this script.

3.4 Virus expression visualization

Similarly to what is done to register and visualize probe tracks, the software can help the user to visualize the virus in the brain.

3.4.1 Registering the virus and navigating the atlas

This file runs based on the images saved from the pre processing step. You can then select the corresponding brain slice with the Waxholm atlas that is available. Make sure you have the right path to the preprocessed images obtained from the previous script, and to the other relevant folders. Firstly, register the virus expression by running the class file `virus_registration` and follow the steps below.

- Always load your atlas first

```
1 register_virus = VirusRegistration(atlas ,  
2 processedhistologyfolder='path')
```

- Add the path here:

```
processed_histology_folder =  
Path('/Users/.../histology/processed')
```

- There are two figures popping up in the screen and to perfectly overlay your histology image with the atlas you need to click 't' to activate the overlay points.
- Choose at-least 4 points in the same order for both the histology slice and the atlas image that is on your computer screen. The numbering will help you remember the clicked points.
- Press 'h' to match the histology with the atlas.
- Press 'o' to visualize the regions boundaries of the Waxholm atlas.
- After having pressed 'h' and 'o' it is possible to either visualize the regions colors by pressing 'v' or start registering the probes by pressing 'r'.
- click the points where you can see the virus expression.
- Press 'c' to delete the virus points.
- Press 'e' to save the registered slice of histology with the virus. Add a name for the file.
- If you have multiple images, once you finish registering the electrode track of one histology image, you'll find the next one lined up on your desktop screen. Follow the same procedure as you did for the first histology image.
- For more options follow the instructions on the console.

NOTE: when interacting with keyboard make sure that one figure between Atlas viewer, Histology viewer or Histology adapted to atlas is selected otherwise the the software will not interact with the keyboard.

3.4.2 Visualizing virus in the 3D atlas

Now you have saved and registered points of the virus. You can further visualize the virus in 3D by running this script

```
1 vis3d_virus = vis_registered_virus(atlas ,  
2 probe_folder)  
3 vis3d_virus.vis2d()  
4 vis3d_virus.vis3d()
```

This script gives you the information about the region and location coordinates of the virus points. Additionally, it creates `info.txt` file where the information are saved. Close the console every time you run this script. Figure 7 shows the 3D visualization of the virus expression in the Waxholm atlas.

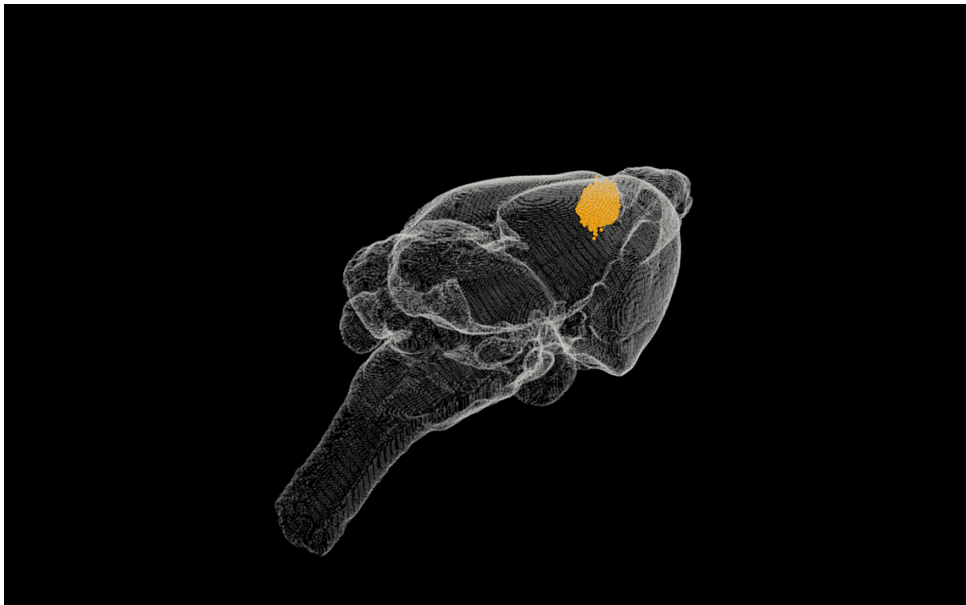


Figure 7: 3D representation of the virus.