Universal Probe Finder User Manual

Version 1.0, 20220518

By Dr. Jorrit Montijn

Cortical Structure and Function group

Netherlands Institute for Neuroscience

E-mail: j.montijn@nin.knaw.nl

Contents

Overview

The Universal Probe Finder is a multi-species probe alignment program that allows you to use neurophysiological markers to better fine-tune your probe's location in the brain. It can use multiple atlases and calculates the stimulus responsiveness of your clusters with the zetatest using only an array of event-onset times.

At this time, the Universal Probe Finder supports the following atlases out-of-the-box:
a. Sprague Dawley rat brain atlas, downloadable at: https://www.nitrc.org/projects/whs-sd-atlas
b. Allen CCF mouse brain atlas, downloadable at: http://data.cortexlab.net/allenCCF/
c. CHARM/SARM NMT_v2.0_sym macaque brain atlas:
https://afni.nimh.nih.gov/pub/dist/doc/htmldoc/nonhuman/macaque_tempatl/atlas_charm.html

It is also possible to add your own Atlas by editing the configAtlas.ini file that is created when you first run the ProbeFinder (see the section "Adding an atlas").

Installation instructions

Before you start:

1. For best performance, make sure your GPU supports OpenGL-accelerated graphics, so update your drivers if required.
2. Download the brain atlas you wish to use, for example:
    a. Sprague Dawley rat brain atlas: https://www.nitrc.org/projects/whs-sd-atlas
    b. Allen CCF mouse brain atlas: http://data.cortexlab.net/allenCCF/
    c. CHARM/SARM macaque brain atlas: https://afni.nimh.nih.gov/pub/dist/doc/htmldoc/nonhuman/macaque_tempatl/atlas_charm.html#download
3. Install the atlas if required (i.e., extract the files to raw nifti/.nii)

Installing the Universal Probe Finder:

1. All required external files are included in the repository's subfolders so you only need to clone this repository: https://github.com/JorritMontijn/UniversalProbeFinder
2. Add the main path to matlab and you're done – or, as the English would say: Bob's your uncle!

User guide

Starting the program

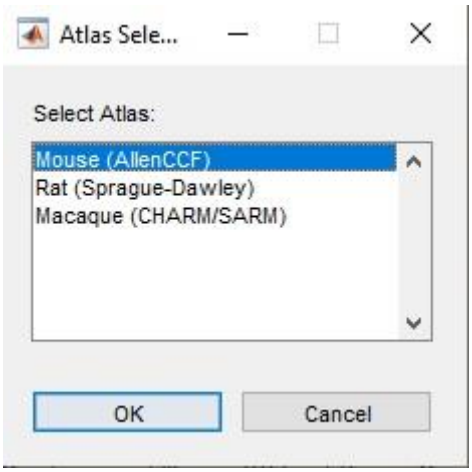
Once you have downloaded the atlas you wish to use (see above) and the Universal Probe Finder repository, it's very easy to start. Open matlab and navigate to where you installed the repository (replace the path with wherever it is on your PC):

```
cd F:\Code\Acquisition\UniversalProbeFinder
```

Next, simply type ProbeFinder, and it will start the program:

```
ProbeFinder
```

You will now be asked to select which atlas you wish to use, which by default is either mouse, rat, or macaque:



If this is the first time you're starting the ProbeFinder, it will now ask you where you installed the atlas you selected. Navigate to its directory and click OK. The ProbeFinder will now load the atlas files, transform it to a standard layout, and start the program.

If you wish, you can now import:

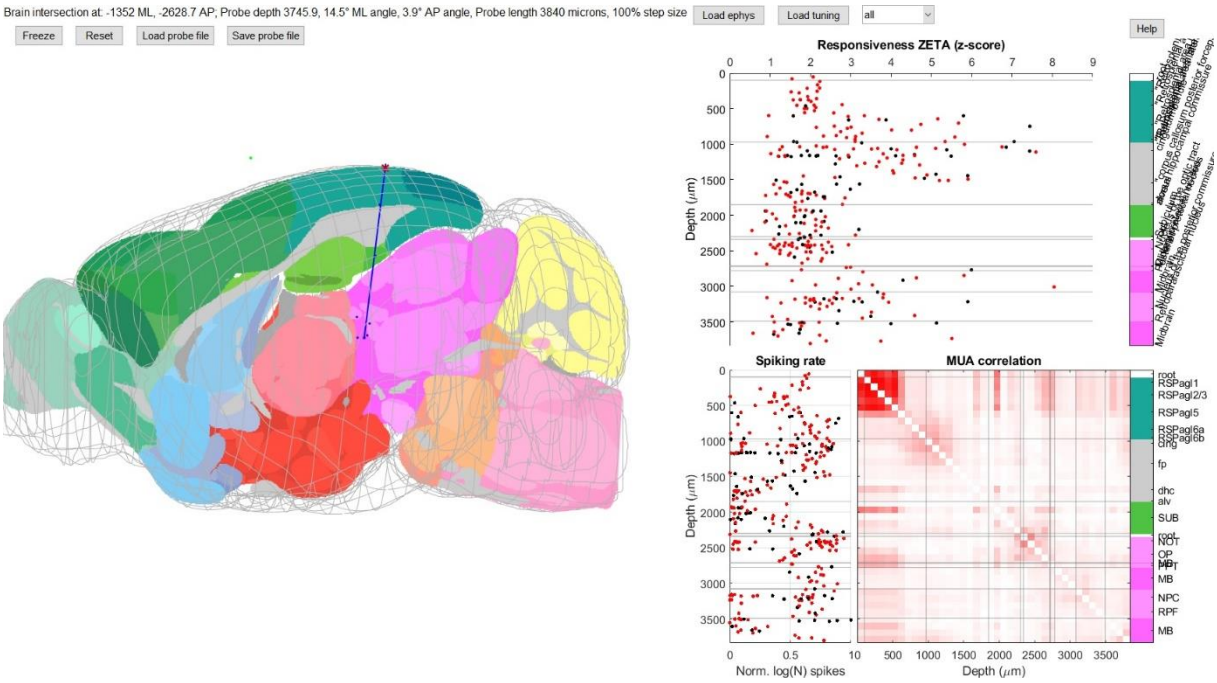
1) A probe file of various formats, such as:
    a. AP_histology output
    b. SHARP-track output
    c. ProbeFinder output
2) Electrophysiological data, such as:
    a. A kilosort 2.5 or kilosort 3 output folder
3) A cluster-responsiveness file, such as:
    a. A matlab file containing event onset times
    b. An Acquipix stimulus log
    c. An Acquipix pre-processed file (synthesis or AP)
    d. A file containing ZETA responsiveness values

If you select option (a) or (b) when loading a responsiveness file, it will compute the responsiveness per cluster based on its spike times and the event onsets you provided (https://elifesciences.org/articles/71969). When it's done, you can save the file for future use.

Using the program

Now that everything is set up, it will show you a pop-up screen with commands. If you ever forget a key combination, you can press F1 or click on the button that says “Help”.

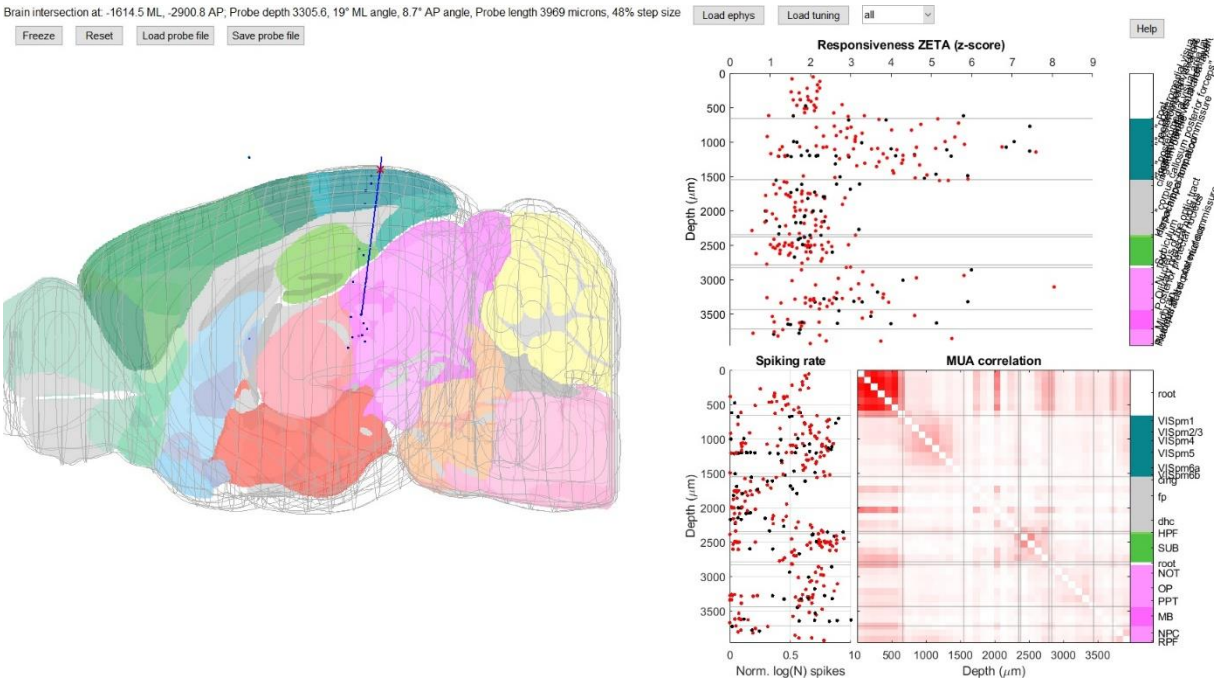
After clicking away the help screen, you will see something like this:



On the left, you can see an outline of the brain and a slice somewhat orthogonal to your viewing direction that goes through the probe. If you imported a probe location file, you will see blue dots that indicate where you thought the probe tracks were located on your histology slices. The green dot floating outside the brain is Bregma.

Normally, this histology-based location is what you would export and use to assign areas to the clusters you recorded on your probe. But often this is not entirely correct! As you can see above, the depth is clearly wrong, but only moving the probe up is not giving a good match either. The spiking rate and the MUA correlation (bottom right) give a pretty good indication of where the top of the cortex is located (MUA correlation is strong for noisy contact points above the brain) and where the white matter bundles lie (clusters show low firing rate). The exact boundaries between different cortical and subcortical areas are more difficult to determine, however.

In this experiment, I showed drifting gratings and natural movies, so calculating the responsiveness ZETA scores using their onset times shows a clear delineation of visually-responsive and non-visually responsive regions (top right). Tweaking the probe location, length and angle a bit, I ended up with this alignment (not perfect yet, but clearly a lot better than before):



Notice how the subiculum has a high MUA correlation, the NOT has a couple of highly visually-responsive cells, and the boundaries of the visual cortex are very sharp when looking at the responsiveness ZETA.

When you're happy with your alignment, you can click on "save probe file" and export the current location. The ProbeFinder will then calculate for each cluster what its area in the brain is, according to your atlas.

Format of the output file

The current probe location is saved in the field "`sProbeAdjusted`" of the structure "`sProbeCoords`". The `sProbeAdjusted` structure contains the following fields, of which the more useful fields are marked in bold:

- probe_vector_cart [2 x 3]: cartesian tip and base coordinates in [ML,AP,DV] atlas space
- probe_vector_sph [1 x 6]: spherical coordinates: [ML AP DV deg-ML deg-AP length]
- probe_vector_intersect [1 x 3]: coordinates of brain entry
- probe_vector_bregma [1 x 6]: bregma-origin format: [ML AP ML-deg AP-deg depth length]
- probe_area_ids_per_depth [1 x N]: vector of area IDs per point along the probe
- probe_area_labels_per_depth [N x 1]: cell array of acronyms per point
- probe_area_full_per_depth [N x 1]: cell array of full names per point
- probe_area_boundaries [1 x P+1]: locations along the probe of area boundaries
- probe_area_centers [1 x P]: centers of areas
- probe_area_ids [1 x P]: id per area of the above
- probe_area_labels [P x 1]: acronym per area
- probe_area_full [P x 1]: full name per area
- *(…) continues on next page*

- **stereo_coordinates:** table with the probe's location in stereotactic coordinates:

Name:	Content:	Unit
ML	Probe's brain entry location along ML (medial-lateral) axis, relative to the atlas origin (e.g., bregma)	Microns
AP	Probe's brain entry location along AP (anterior-posterior) axis, relative to the atlas origin (e.g., bregma)	Microns
Angle_ML	Angle of the probe in the ML direction (see below)	Degrees
Angle_AP	Angle of the probe in the AP direction (see below)	Degrees
Depth	Depth below the brain entry of the lowest recording channel of the probe (i.e., usually the depth of the tip)	Microns
Probe_Length	Length of the probe after stretching/shrinking	Microns

- **probe_area_ids_per_cluster**: area IDs per cluster
- **probe_area_labels_per_cluster**: area acronyms per cluster
- **probe_area_full_per_cluster**: full name of the area per cluster

Note that these final three *_per_cluster* variables only have values if electrophysiological data was loaded when the probe file was saved.

Definition of ML/AP angles

Defining a single axis of 360-degree rotation in R^3 is trivial and defines a unique set of points. Unfortunately, combining two 360-degree axes of rotation in 3-dimensional space leads to a non-unique spherical projection space, where multiple combinations of the two angles describe the same point. Describing these problems in detail is beyond the scope of this manual (see https://en.wikipedia.org/wiki/Spherical_coordinate_system#Unique_coordinates for more details), but it is important to note that converting between cartesian and spherical coordinate systems can be tricky.

For our purposes of defining a probe's location, it is most useful to have two axes that align more or less with our two most important axes: ML and AP. Moreover, a natural "origin" for the probe would be to define the direction going straight down as 0 degree ML and 0 degree AP. This requires rotating the spherical coordinate system (fig. 1).

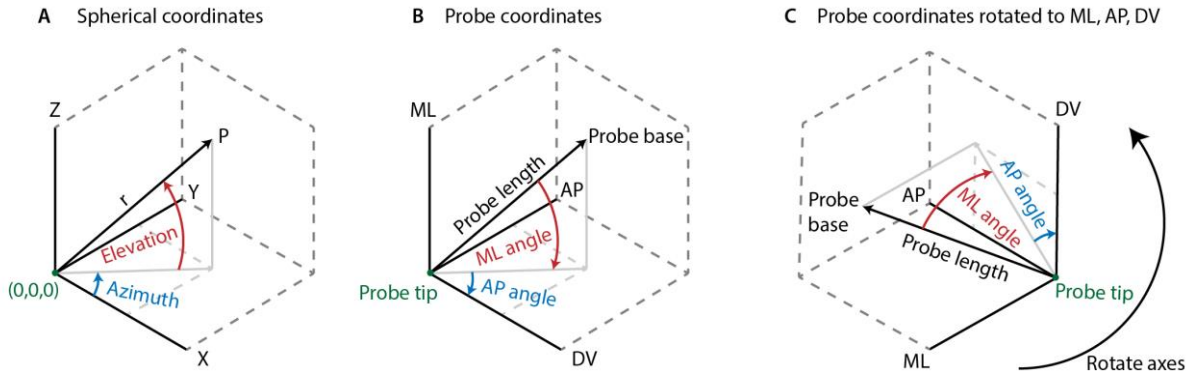


Figure 1. Definition of ML and AP angles. A) Standard spherical coordinate system using elevation and azimuth. Azimuth is defined on the interval [-π, π] and elevation on [-π/2, π/2]. B) X,Y,Z, replaces by stereotactic axes. Notice the change in angle direction. C) Rotated axes aligned with atlas directions.

Furthermore, to ensure that the necessary discontinuities in the rotations occur when the probe is upside-down (which presumably is not a common recording orientation), rather than close to its "origin" position, we invert the AP-axis and apply the following transformations:

```
    if dblAngleAP < -90 && dblAngleML > 0
        dblAngleAP = dblAngleAP + 180;
        dblAngleML = -dblAngleML + 180;
    elseif dblAngleAP < -90 && dblAngleML < 0
        dblAngleAP = dblAngleAP + 180;
        dblAngleML = -dblAngleML - 180;
    elseif dblAngleAP > 90 && dblAngleML > 0
        dblAngleAP = dblAngleAP - 180;
        dblAngleML = -dblAngleML + 180;
    elseif dblAngleAP > 90 && dblAngleML < 0
        dblAngleAP = dblAngleAP - 180;
        dblAngleML = -dblAngleML - 180;
    end
```

The specific implementations can be found in the functions `PH_CartVec2SphVec`, `PH_SphVec2CartVec`, `PH_BregmaVec2SphVec`, and `PH_SphVec2BregmaVec`.

Adding an atlas

The ProbeFinder reads a configuration .ini file to find which atlases are installed. If no configAtlas.ini file is present, it will create a default file containing metadata on the Allen Brain mouse atlas, Sprague-Dawley rat atlas, and CHARM/SARM macaque atlas. If you wish to add an atlas, you can edit the .ini file by adding another atlas entry set. For example, the first entry looks like this:

```
[sAtlasParams(1)]
name='Mouse (AllenCCF)'
pathvar='strAllenCCFPath'
loader='AL_PrepABA'
downsample=2
```

It specifies:

- name: the name of the atlas,
- pathvar: the name of path variable that is saved in configPF.ini (the atlas's path location)
- loader: the name of the function that pre-processes the atlas files (see below)
- downsample: the amount of downsampling when plotting an atlas slice

The name, pathvar, and downsample are self-explanatory, but the loader requires some more explanation. The syntax of a loader function is:

```
sAtlas = name_of_loader_function(str_atlasname_Path)
```

The loader function reads the atlas files at the specified path and outputs a structure sAtlas with the following fields:

Field name	Size	Description
.av	[ML x AP x DV]	Annotated volume, where a value in av denotes an area ID that can directly index into .st; i.e., sAtlas.st(sAtlas.av(x,y,z)).name gives the full name of the area at location x,y,z.
.tv	[ML x AP x DV]	Grey-scale template volume in range [0 255]
.st	[N x T] table	N-entry table, where N is highest area index in .av
.Bregma	[1 x 3]	Origin of atlas in native atlas voxel coordinates [ML AP DV]
.VoxelSize	[1 x 3]	Size of a single voxel in microns [ML AP DV] Note: currently only isometric voxels are supported
.BrainMesh	[P x 3]	Mesh of brain outline, where each vertex is a [1 x 3] point, and curve-ends are denotes by a [nan nan nan] entry. Brain meshes can be created from a .av or .tv volume using `getTrace3D`
.Colormap	[N x 3]	N-entry RGB color map, specifying a color for each area in .av
.Type	string	Name, e.g.: Allen-CCF-Mouse

The area-table .st must have least the following fields:

Field name	Description
st.id	ID of area (numeric)
st.name	Full name of area (string)
st.acronym	Short name of area (string)
st.parent_structure_id	ID of parent area

If you wish to add an atlas to the program, we are happy to help out. You can contact us by e-mail or via the github repository.

Acknowledgements

This work is based on earlier work by people from the cortex lab, most notably Philip Shamash and Andy Peters. See for example: https://www.biorxiv.org/content/10.1101/447995v1
This repository includes various functions that come from other repositories, credit for these functions go to their creators:
https://github.com/petersaj/AP_histology
https://github.com/JorritMontijn/Acquipix
https://github.com/JorritMontijn/GeneralAnalysis
https://github.com/kwikteam/npy-matlab
https://github.com/cortex-lab/spikes
https://github.com/JorritMontijn/zetatest

Troubleshooting

Question ("actually, it's more of a comment"): *It doesn't work*

Answer: Restart your PC, make sure the UniversalProbeFinder is on your matlab path, and try again.

Q: *Why is it so slow?*

A: You're probably not using OpenGL rendering. If you're from the future, your matlab version might have broken the OpenGL rendering switch. If so, please file a bug report. If you're using anything that's R2022a or earlier, you might need to update your GPU or your graphics drivers.

Q: *I'm using R2022a or earlier, updated my drivers, have a compatible GPU, and it's still slow.*

A: It's possible that the atlas slices are slowing things down. If hiding the slices with "s" speeds things up, you might want to consider editing the configAtlas.ini file to increase the downsampling for your atlas (Note: values are rounded to the nearest integer). For example, if you're using the Allen Brain atlas, you will see:

```
[sAtlasParams(1)]
name='Mouse (AllenCCF)'
pathvar='strAllenCCFPath'
loader='AL_PrepABA'
downsample=2
```

You can change the line "`downsample=2`" to (e.g.) "`downsample=5`", restart the ProbeFinder, and see if it makes a difference.

Q: *I found a bug*

A: If you've fixed it, you can make a pull request, otherwise you can create a bug report here: https://github.com/JorritMontijn/UniversalProbeFinder/issues. Please copy/paste the error message and provide as much detail as you can about what you were doing when it happened. If I cannot recreate the issue, I probably won't be able to fix it.