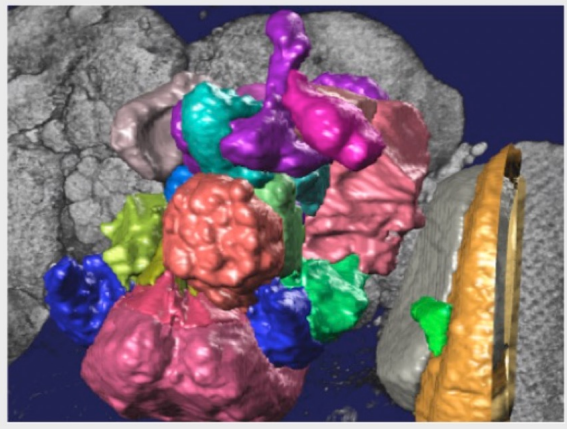


Extend Your Research with Vaa3D Plugins

Yufeng Liu

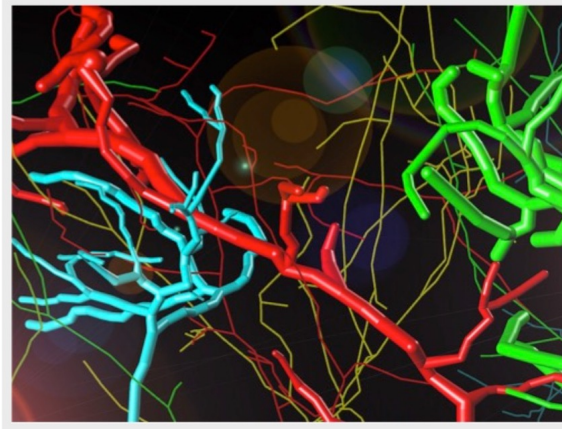


Vaa3D: A Swiss Army knife for exploring big big image data



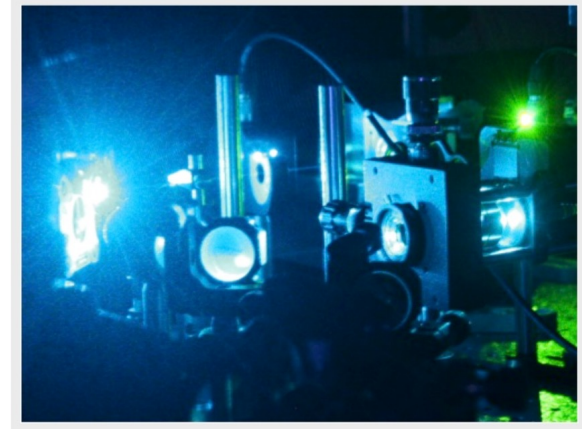
FAST

Vaa3D visualizes and explores big 3D/4D/5D images with giga-voxels and even tera-voxels, within seconds or sub-seconds!



COOL

Vaa3D extracts complex surface objects from images, and performs comprehensive analyses such as brain connectome mapping.



EXTENSIBLE

100+ plugins for image acquisition, microsurgery, data management and analysis, and massive-scale pipelining.

V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets

Hanchuan Peng^{*}, Zongcai Ruan, Fuhui Long, Julie H. Simpson, and Eugene W. Myers

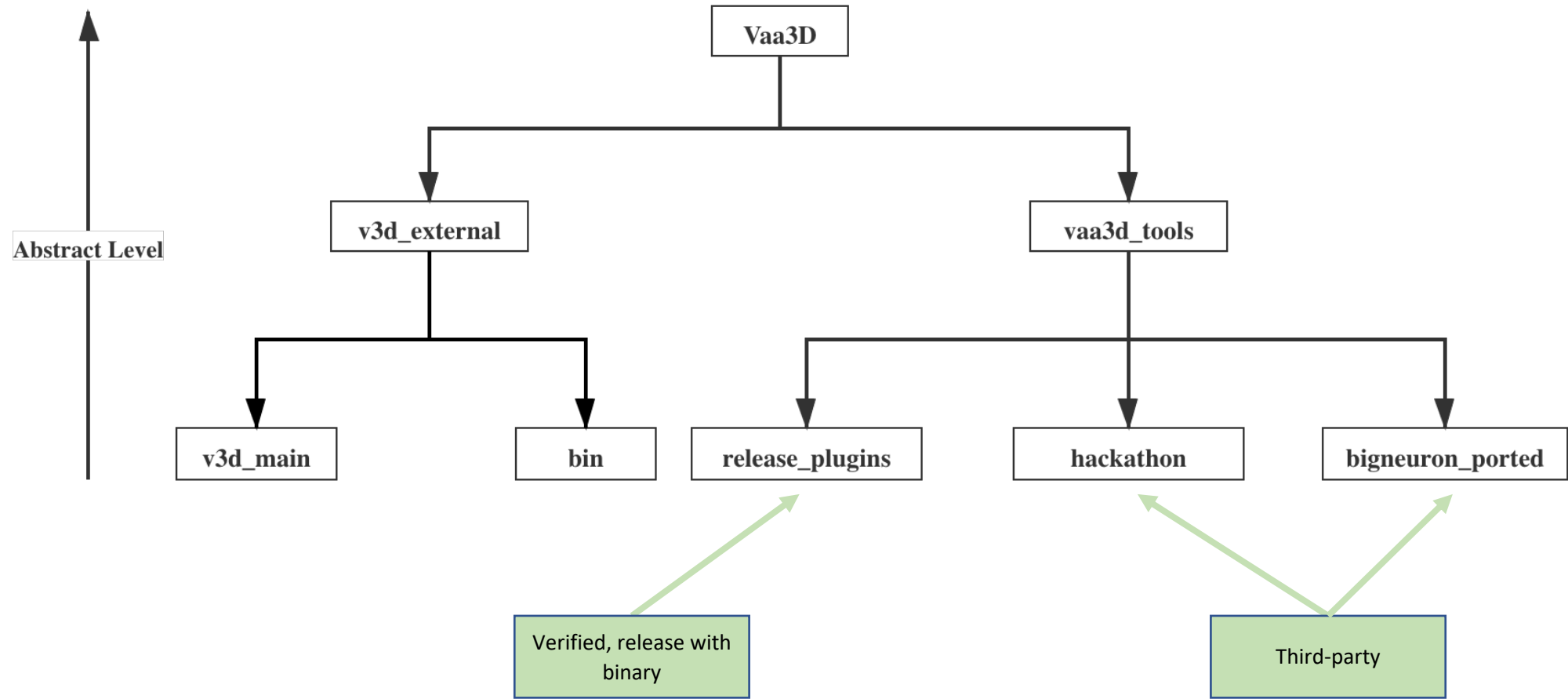


Extensibility: Plugins

- ❑ Dynamic libraries on top of Vaa3D that empower the main program with specific function
 - ❑ make use of Vaa3D functions
 - ❑ call other plugin functions
- ❑ Two-folds
 - ❑ Large number of available plugins
 - ❑ Plugin implementation required only minimal effort

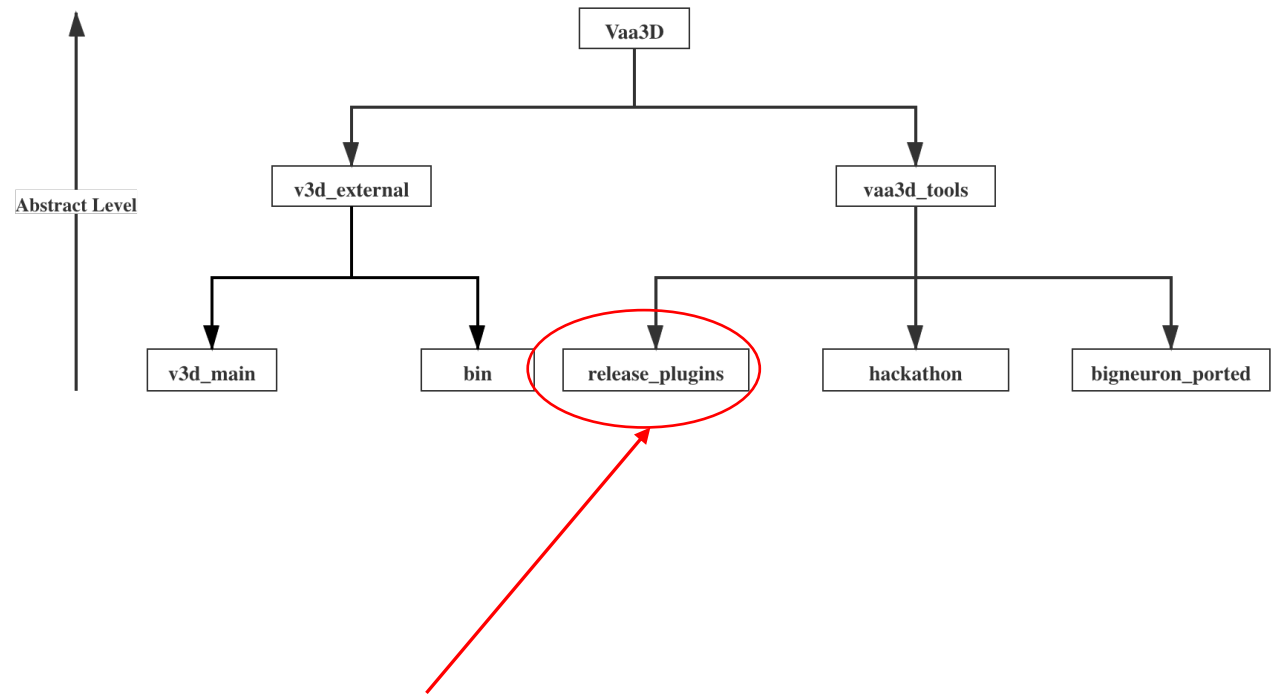


Vaa3D architecture



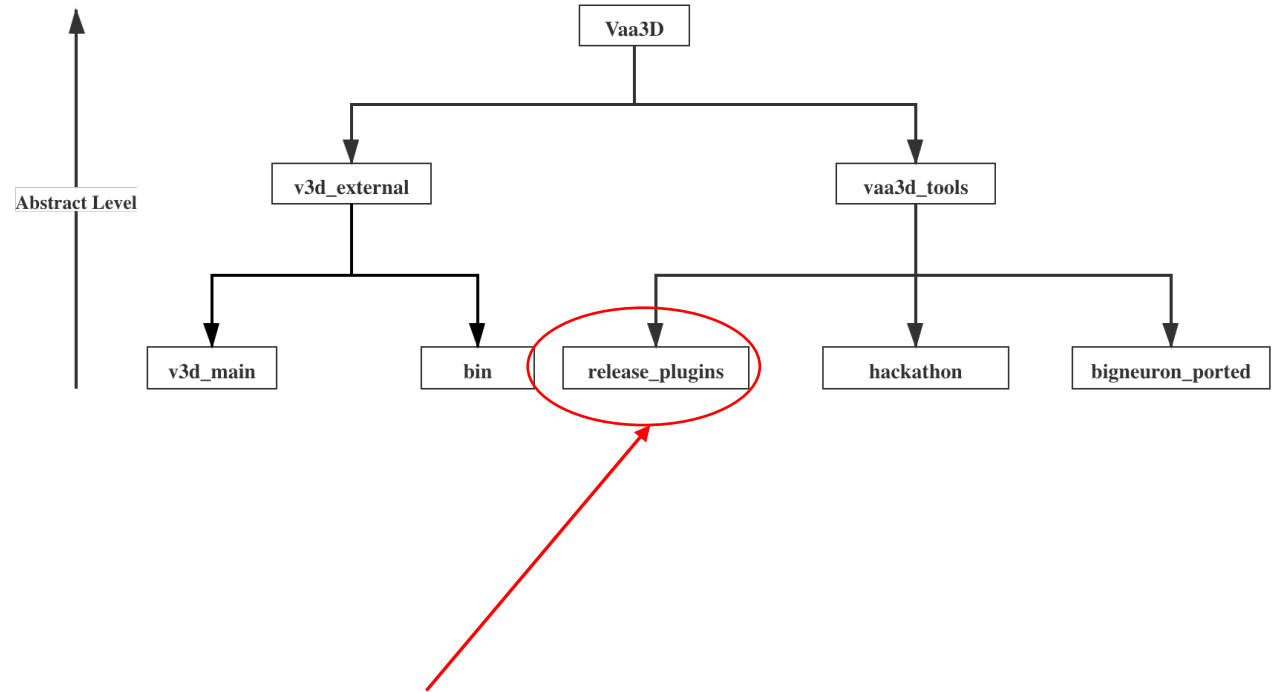
Available plugins: Built-in plugins

- ❑ 140+ plugins
- ❑ pre-built with binary, automatic built while compiling
- ❑ source code located at `released_plugins``
- ❑ binaries located at `bin``
- ❑ full list: [Appendix](#)



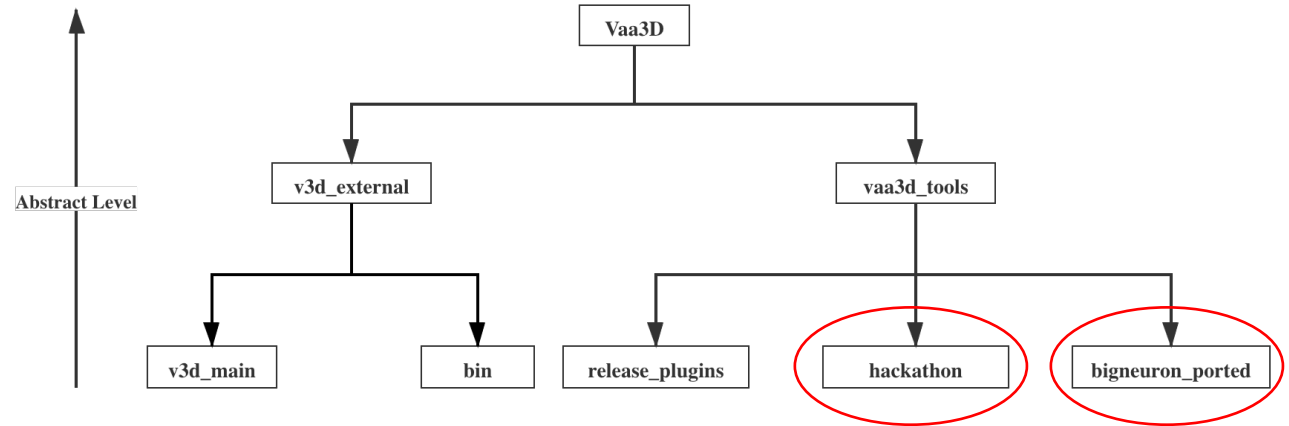
Available plugins: Built-in plugins

- ❑ Diverse functionalities:
 - ❑ 27 automatic tracing algorithms
 - ❑ Neuron analysis, resampling
 - ❑ Image analysis, transformation, filtering, visualization
 - ❑ Registration, stitching
 - ❑ File IO and conversion
 - ❑ ...

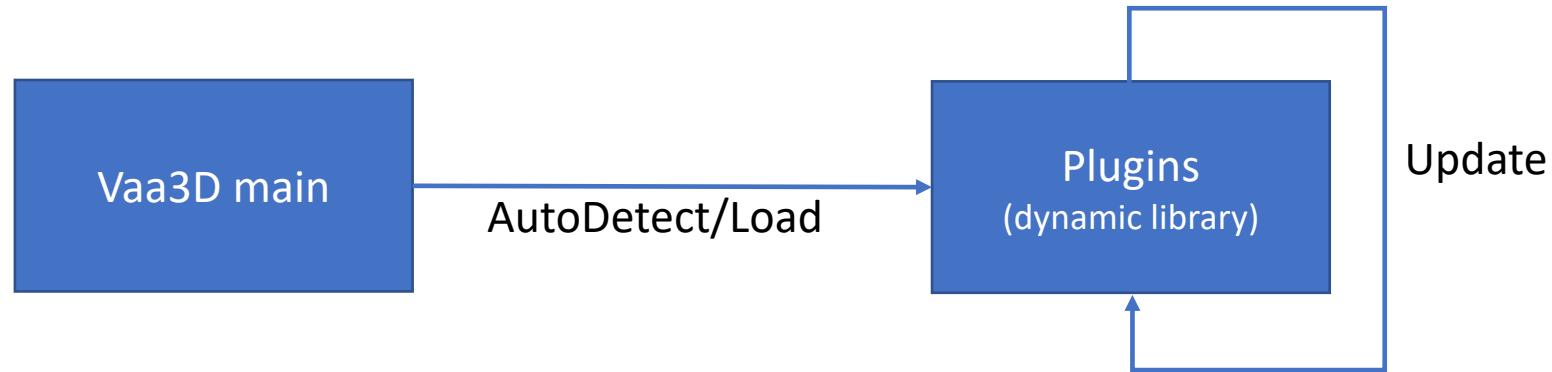


Available plugins: Third-party

- ❑ Hackathon & BigNeuron_ported
 - ❑ ~400 plugins!!!
 - ❑ Compile manually
 - ❑ Users are encouraged to implement their plugins here.



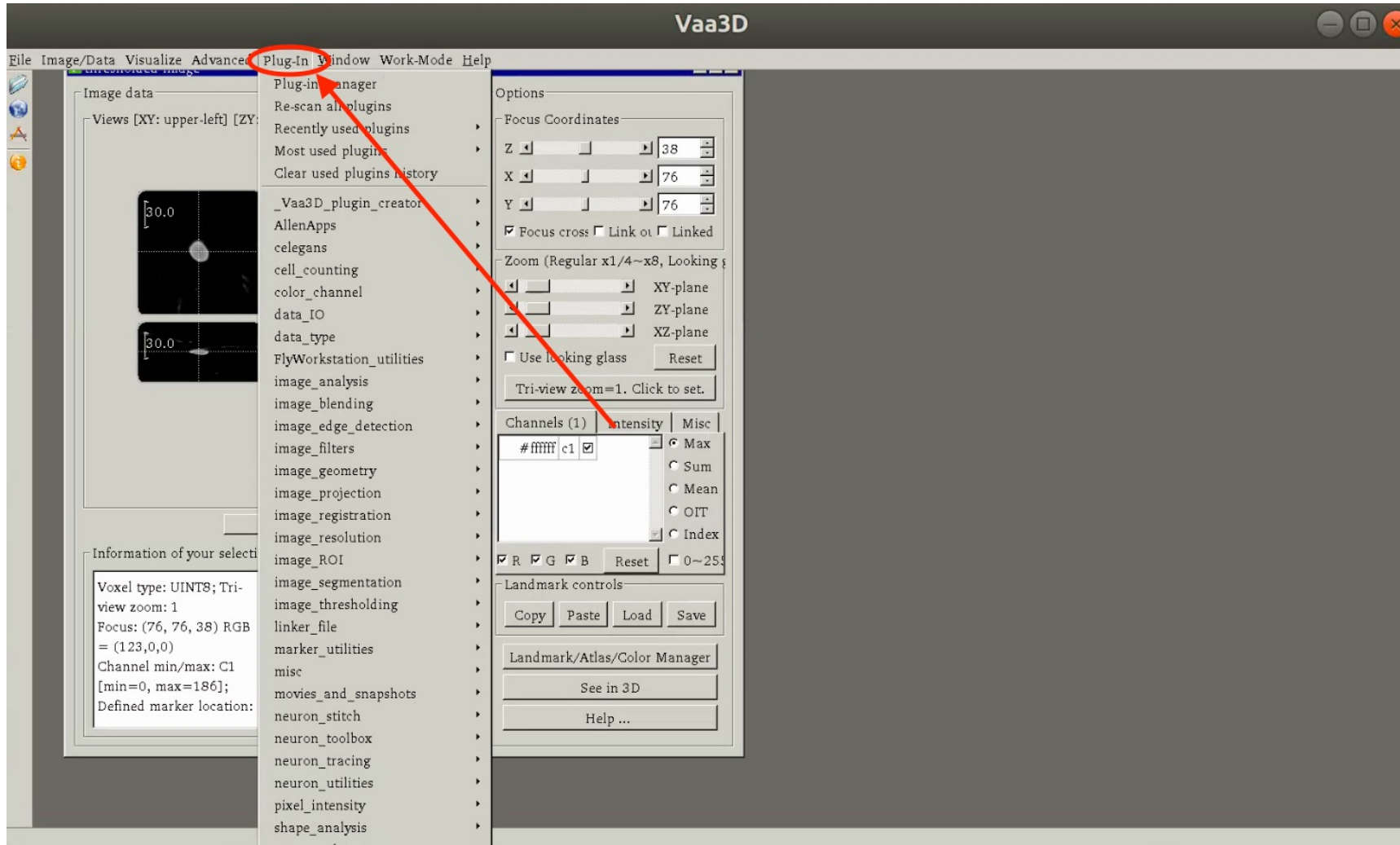
High Extensibility with Minimal Effort



- ❑ Plugins are independent with main program
 - ❑ Interact at dynamic library level (*.so/*.dll/*.dylib)
 - ❑ Addition/modification of plugin does not require re-compiling of main program
- ❑ Vaa3D provide facilities that
 - ❑ automatically detect, load, and call plugins
 - ❑ reserves extensible interface for new plugins



Usage of plugins: through main menu



Usage of plugins: through main menu



Usage of plugins: through command line

- large scale data
- Additional configurable parameters for some plugins
- Better exception control
- Speed up via parallelization



Usage of plugins: through command line

Full list of plugins:

```
vaa3d -h (for Mac OS and Linux)  
vaa3d_msvc.exe /h (for Windows)
```

Help information of a specific plugin:

```
vaa3d -h -x <plugin_name> (for Mac OS and Linux)  
vaa3d_msvc.exe /h /x <plugin_name> (for Linux)
```



Usage of plugins: through command line

In Linux shell:

```
vaa3d -h -x <plugin_name> #find out the usage
vaa3d -x <app2_so_path> -f app2 -i <input_image> \
  -o <output_image> -p <marker_file> 0 AUTO 0 \
  # execute APP2 in auto mode, with pre-defined soma location
```

Through other languages, e.g. python:

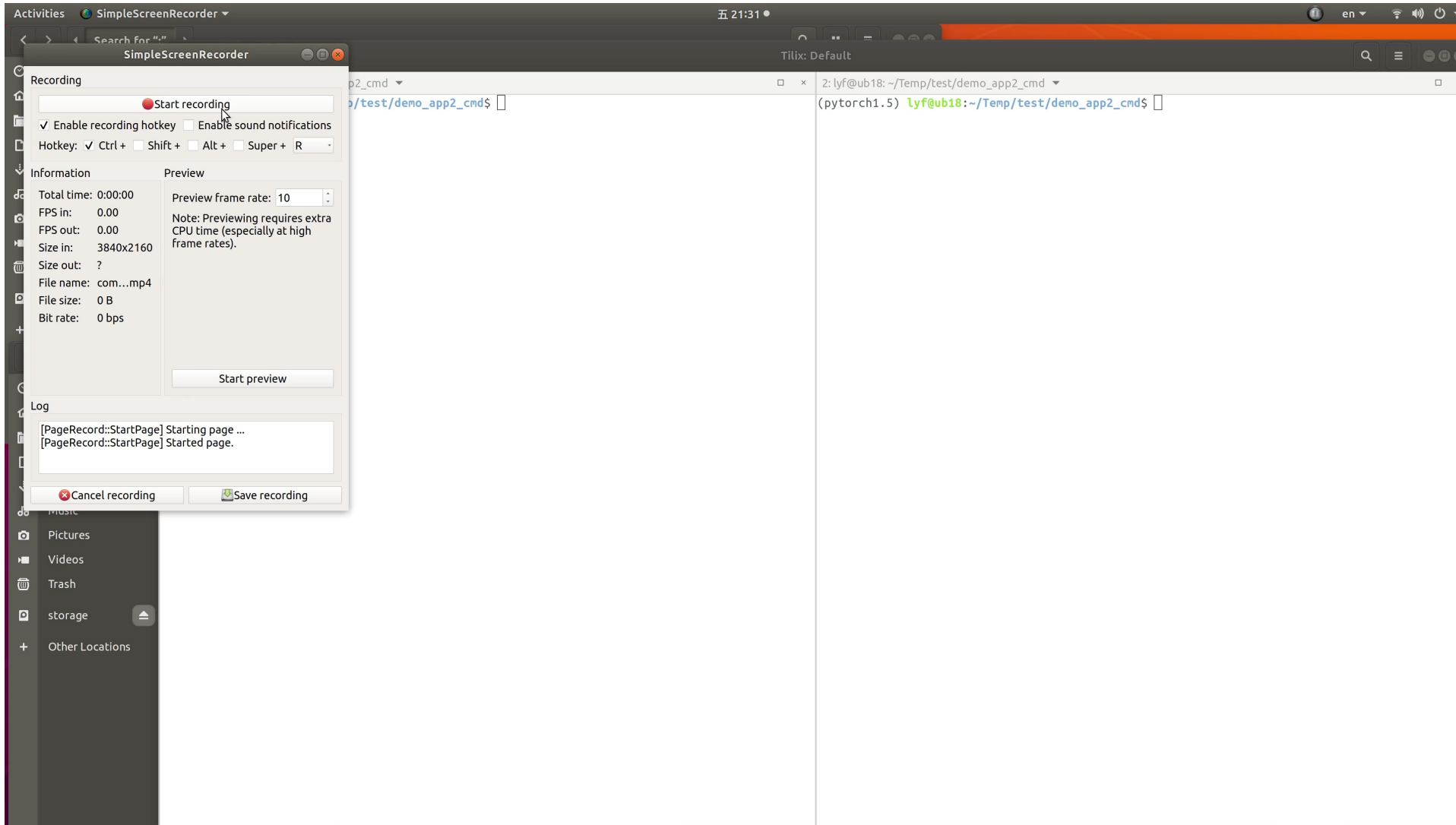
```
def exec_app2(input_image, output_image,
              vaa3d='<vaa3d_external>/bin/vaa3d',
              plugin='<vaa3d_external>/bin/plugins/neuron_tracing/Vaa3D_Neuron2/libvn2.so',
              marker_file=<your_marker_file>):

    cmd_str = f'{vaa3d} -x {plugin} -f app2 -i
               {input_image} -o {output_image}
               -p {marker_file} 0 AUTO 0'

    try:
        p = subprocess.check_output(cmd_str, timeout=6000, shell=True)
    except subprocess.TimeoutExpired:
        print(f'Execution of image: {input_image} is too time-consuming. Skip!')
        p = ''
    return p
```



Usage of plugins: through command line



The screenshot displays the SimpleScreenRecorder application window. The main window is titled "SimpleScreenRecorder" and contains several sections:

- Recording:** A "Start recording" button with a red dot indicator. Below it are checkboxes for "Enable recording hotkey" (checked) and "Enable sound notifications" (unchecked). A hotkey dropdown menu shows "Ctrl + Shift + R".
- Information:** A table of recording statistics:

| | |
|-------------|-----------|
| Total time: | 0:00:00 |
| FPS in: | 0.00 |
| FPS out: | 0.00 |
| Size in: | 3840x2160 |
| Size out: | ? |
| File name: | com...mp4 |
| File size: | 0 B |
| Bit rate: | 0 bps |
- Preview:** A "Preview frame rate" dropdown set to "10". A note states: "Note: Previewing requires extra CPU time (especially at high frame rates)." A "Start preview" button is located below.
- Log:** A text area containing the following log entries:

```
[PageRecord::StartPage] Starting page ...  
[PageRecord::StartPage] Started page.
```
- Buttons:** "Cancel recording" (with a red X icon) and "Save recording" (with a floppy disk icon) buttons are at the bottom.

In the background, a terminal window titled "Tilix: Default" is open, showing a shell prompt and the execution of a command: `(python3.8) lyf@ub18: ~/Temp/test/demo_app2_cmd$`. The terminal output shows the command being executed in a virtual environment: `(pytorch1.5) lyf@ub18: ~/Temp/test/demo_app2_cmd$`.



Write your own plugin: pre-requisite

Environmental pre-requisite:

- Proper Qt version installed
- Vaa3D source code downloaded (<http://vaa3d.org>)
- C++ compiler (e.g. gcc)

More informations:

- Supported versions refer to: https://github.com/Vaa3D/Vaa3D_Wiki/wiki



Write your own plugin

Structure of plugin:

- plugin.h
- plugin.cpp
- plugin.pro

Plugin creator:

- `_Vaa3D_plugin_creator_` plugin, GUI generate plugin template automatically

More informations:

- Guidelines: https://github.com/Vaa3D/Vaa3D_Wiki/wiki/PluginDesignGuide.wiki



Write your own plugin: an example

The screenshot displays a Linux desktop environment. In the foreground, the SimpleScreenRecorder application window is open, showing a 'Recording' dialog. The dialog includes a 'Start recording' button, checkboxes for 'Enable recording hotkey' (checked) and 'Enable sound notifications', and a hotkey selection menu currently set to 'Ctrl + R'. Below this, there are 'Information' and 'Preview' sections. The 'Information' section shows recording statistics: Total time: 0:00:00, FPS in: 0.00, FPS out: 0.00, Size in: 3840x2160, Size out: ?, File name: wri...mp4, File size: 0 B, and Bit rate: 0 bps. The 'Preview' section shows a 'Preview frame rate' of 10 and a note: 'Note: Previewing requires extra CPU time (especially at high frame rates)'. At the bottom of the dialog, there is a 'Start preview' button and a 'Log' section containing the text: '[PageRecord::StartPage] Starting page ...' and '[PageRecord::StartPage] Started page.'. At the very bottom of the dialog are 'Cancel recording' and 'Save recording' buttons.

In the background, a terminal window titled 'Tilix: Default' is open, showing a shell prompt at the directory `~/Softwares/installation/Vaa3D/vaa3d_tools`. The terminal output shows the command `released_plugins$` and the prompt `lyf@ub18:~/Softwares/installation/Vaa3D/vaa3d_tools$`.

