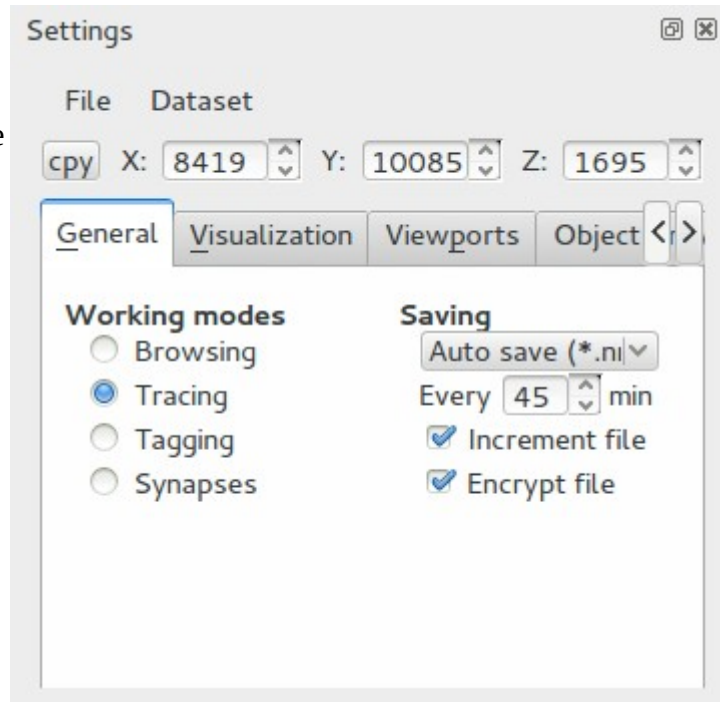


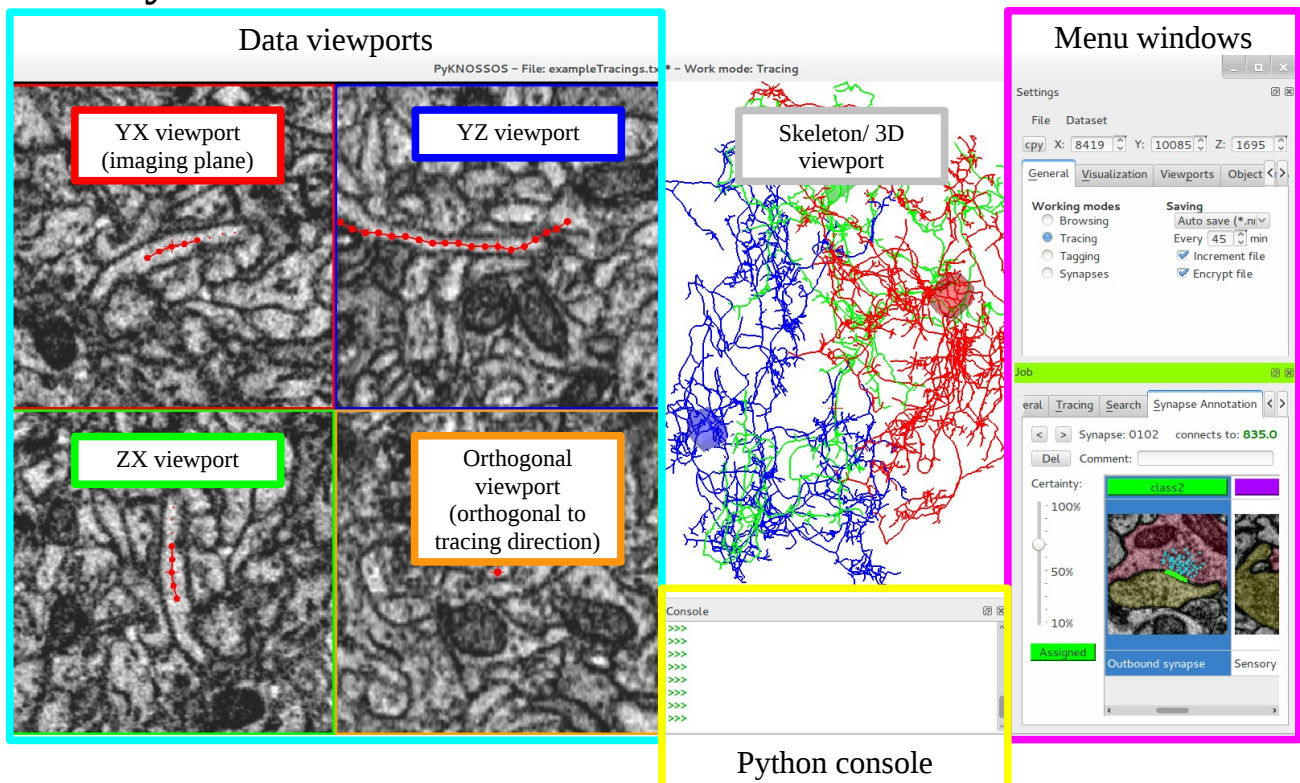
PyKnossos Manual Version 1.3

Getting Started

1. Start PyKnossos by double-clicking on "PyKnossos.exe"
2. Load a new cubed dataset:
 1. Click on the *Dataset* menu in the *Settings* window on the top right
 2. Choose "Load new dataset"
 3. Browse to desired dataset and open the configuration file "PyKnossos.conf"
4. If the **current position** is inside of your dataset the data around that position will be loaded.
NOTE: The position is given units of pixels
3. Open an existing skeleton file:
Click *Open* in the *File* menu in the *Settings* window



The PyKnossos Main Window



- **Menu windows:** Hide/show the *Settings* and *Job* window with the M key
- **Python console:** Python console for command-line control (only available in expert mode).

Data viewports

- YX viewport: Shows the imaging plane
- Orthogonal viewport: Shows the plane perpendicular to the tracing direction or the direction of the current process.
- YZ, ZX viewports: Show the corresponding reslices at the current positions, depending on the reference viewport set in:
Settings window → *Viewports* tab → *Data viewports* → Reference VP
 - YX: The YZ, ZX viewports are perpendicular the *YX viewport*
 - orth.: The two viewports are perpendicular to the to the *Orthogonal viewport*
- Each data viewport is visible in the other viewports as a line with the corresponding frame color. These lines can be hidden by clicking on "Hide crosshairs" in the *Toolbox* tab.
- X key: Shows/hides the red cross in the center that marks the current location/ focal point
- Maximize/minimize viewport: Double-click in the corresponding viewport
- **Panning**: Left click or keep the mouse wheel pressed and move the mouse cursor
- **Move perpendicular** to the currently active viewport:
 - F/D keys
 - Scroll with the mouse wheel
 - UP/DOWN keys
- **Zoom in/out**:
 - **with mouse**: Keep CTRL + right mouse button pressed while moving the mouse cursor up/down OR
 - keyboard: +/- keys
- Reset zoom to 0 (native resolution) Space-key
- **S key**: Jump to the currently selected object
- **Select node/object**: Left click on node/object
- Q key: Switch between multiple datasets, e.g. from compressed to uncompressed cubes.

Skeleton/ 3D viewport

- Shows the skeletons and other objects in 3D space.
- Maximize/minimize viewport: Double-click in the corresponding viewport
- **Rotate**: Keep CTRL + left mouse button pressed while moving the mouse cursor
- **Zoom in/out**: Keep CTRL + right mouse button pressed while moving the mouse cursor up/down
- **Panning**: Left click or keep the mouse wheel pressed and move the mouse cursor
- **J key**: Jump to the current position of the data viewports
- **Select node/object**: Left click on node/object
- X key: Shows/hides the red cross in the center that marks the current location/ focal point

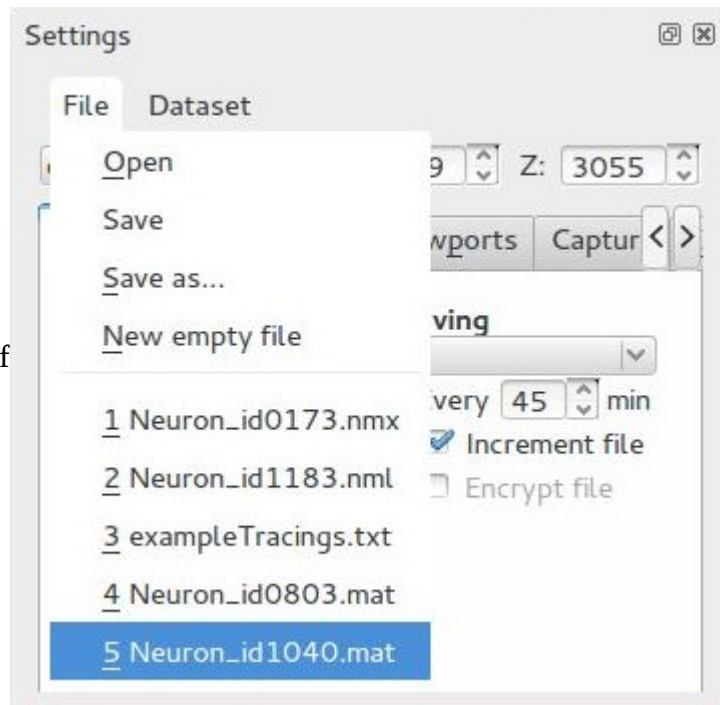
The *Settings* window

Current Position (See red box in the top-right figure on the first page)

- **X/Y/Z:** Coordinates of the current position in units of pixels of the cubed dataset
- **cpy:** Copy current position coordinates into clipboard
- Paste coordinates from clipboard: Click in one of the current position coordinate fields and press CTRL + V keys to insert coordinates from the clipboard (e.g. in the format "[8419, 10085, 1695]")

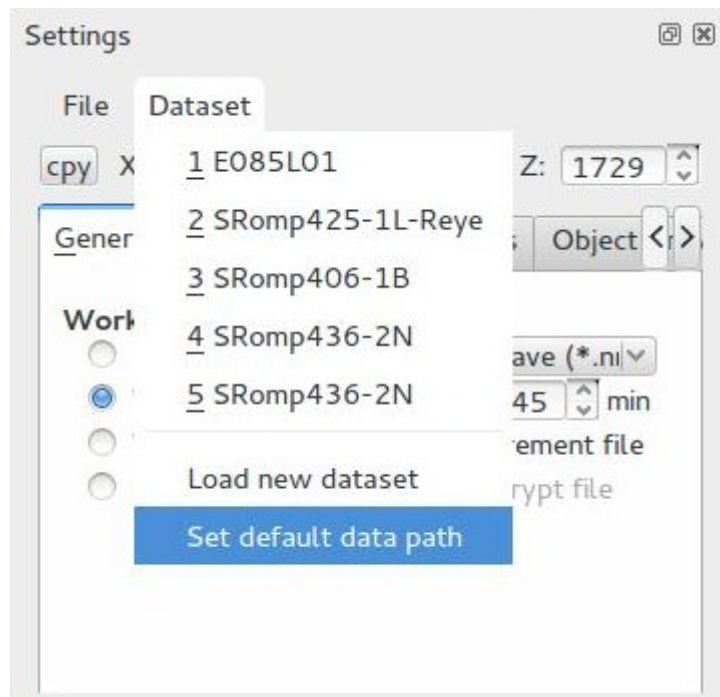
File menu

- **Open:** Load existing skeleton files. Currently the following file formats are supported:
 - *.nmx: Compressed, native file format of PyKnossos (default)
 - *.nml: Native file format of Knossos (XML-based)
 - *.txt: Text file containing a list of file paths of any supported file format
 - *.mat: MATLAB file (only available in expert mode)
- **Save/Save as:** Save the current skeleton in the *.nmx file format.
- **New empty file:** Create a new empty file
- **1-5:** History of previously loaded/ files



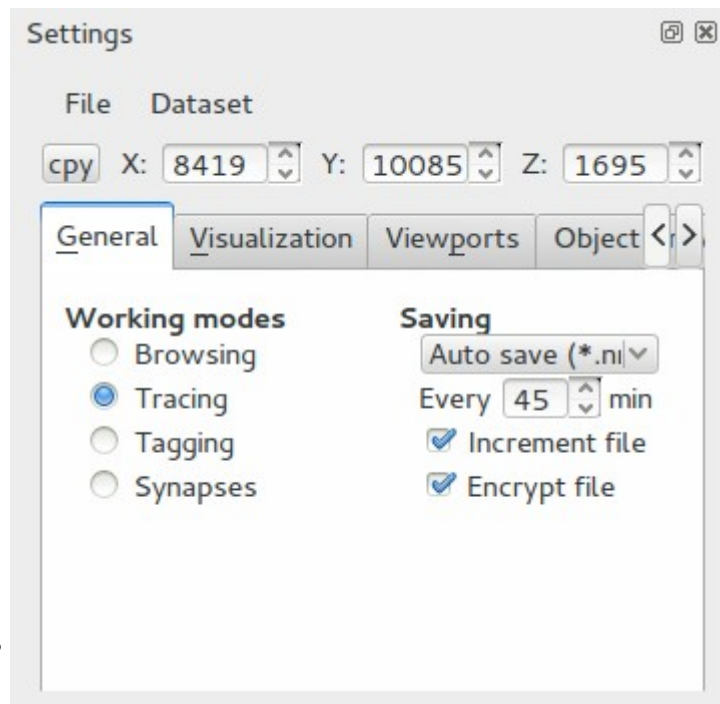
Dataset menu

- **1-5:** History of previously loaded datasets
- **Load new dataset:** Load a new cubed dataset by browsing to the dataset and selecting the corresponding configuration file (usually "PyKnossos.conf")
- Set the default path to the datasets



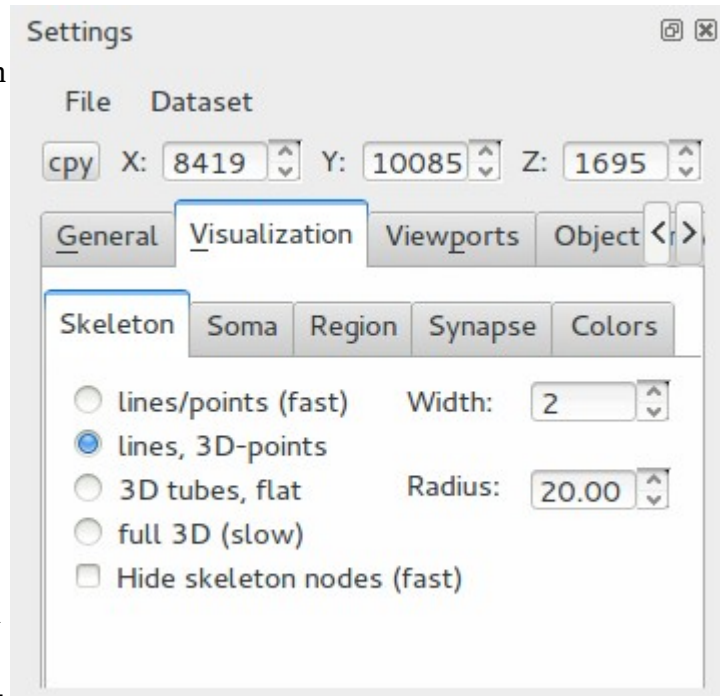
General tab

- **Working modes:**
 - Browsing: Browse through your dataset, select nodes, etc. Modifying skeletons/nodes/tags is not possible.
 - Tracing: Trace/outline objects in the dataset by putting single or edge-connected nodes.
 - Tagging: Tag objects in the data (by putting nodes)
 - Synapses: Special mode for efficient synapse annotation
- **Saving:**
 - Auto save: Defines...
 - ... what happens if you press CTRL + S or click Save in the File Menu
 - ... if you want to auto save (*.nmx) the work every ... minutes. It's recommended to save at least every 5-10min to prevent loss of data and work.
 - Increment file: Increments the file number automatically whenever it is saved.
 - Encrypt file: Save encrypted nmx-files (can only be changed in expert modes)

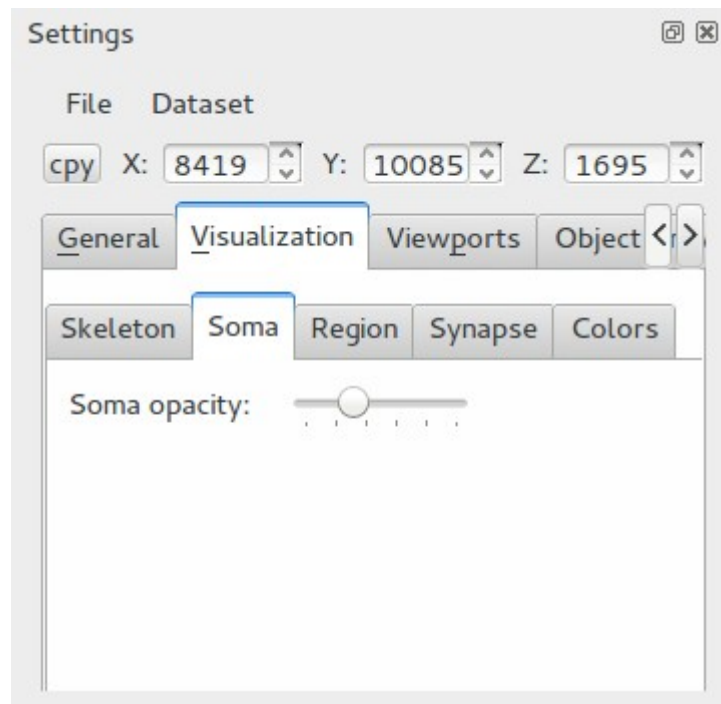


Visualization tab

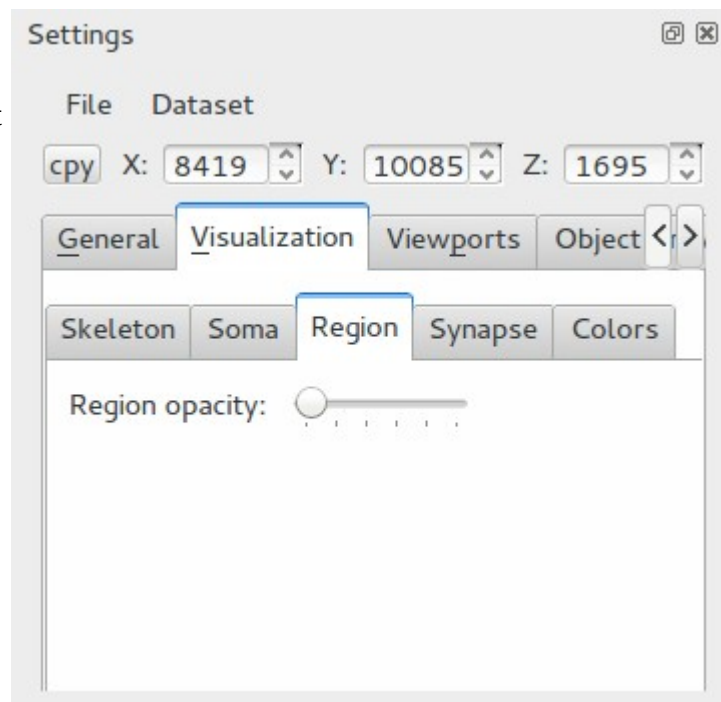
- **Skeleton:**
 - lines/points: Fastest visualization mode. Nodes are visualized as points and edges as lines.
 - lines/3D-points: Nodes are visualized as spheres without light effects.
 - 3D tubes, flat: Edges are visualized as 3D-tubes without light effects
 - full 3D: Nodes/edges are visualized as spheres/tubes with lighting. This is the slowest and most memory demanding visualization option.
 - Width: Setting the point size and line width of edges
 - Radius: Setting the radius of 3D-nodes and tubes
 - Hide skeleton nodes: Hide skeleton nodes for even faster rendering



- **Soma:**
 - Soma opacity: Set the opacity level of somas from transparent to opaque

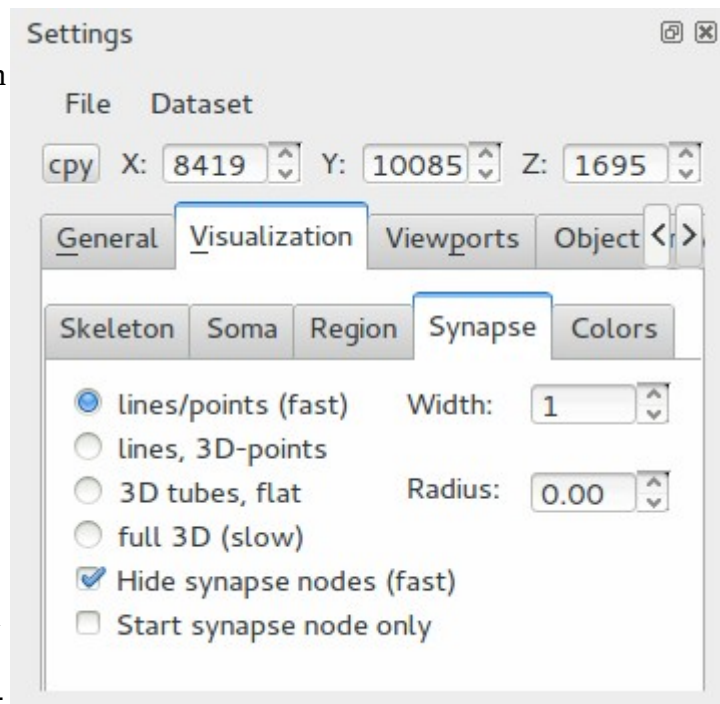


- **Region:**
 - Region opacity: Set the opacity level of regions from transparent to opaque



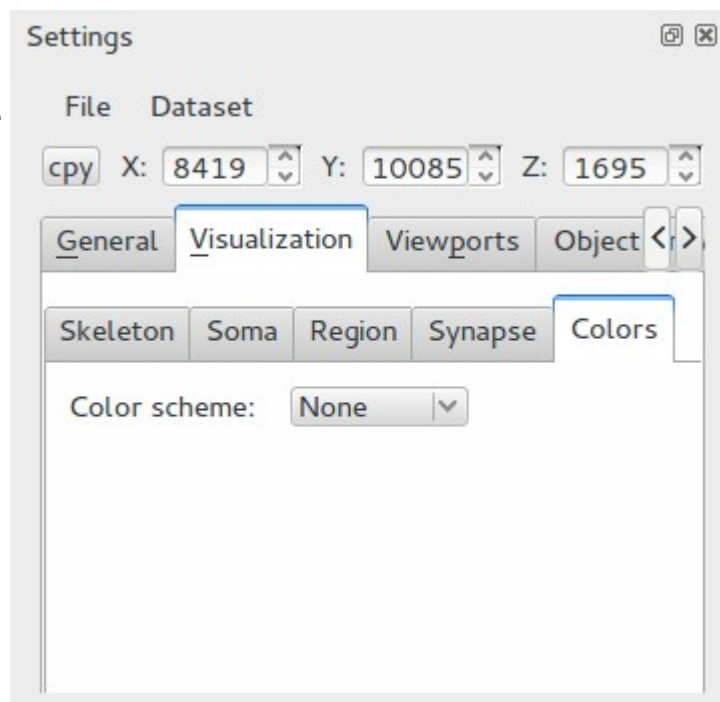
- **Synapse:**

- lines/points: Fastest visualization mode. Nodes are visualized as points and edges as lines.
- lines/3D-points: Nodes are visualized as spheres without light effects.
- 3D tubes, flat: Edges are visualized as 3D-tubes without light effects
- full 3D: Nodes/edges are visualized as spheres/tubes with lighting. This is the slowest and most memory demanding visualization option.
- Width: Setting the point size and line width of edges
- Radius: Setting the radius of 3D-nodes and tubes
- Hide synapse nodes: Hide synapse nodes for even faster rendering
- Start synapse node only: Only show synapse node on pre-synaptic skeleton



- **Colors:**

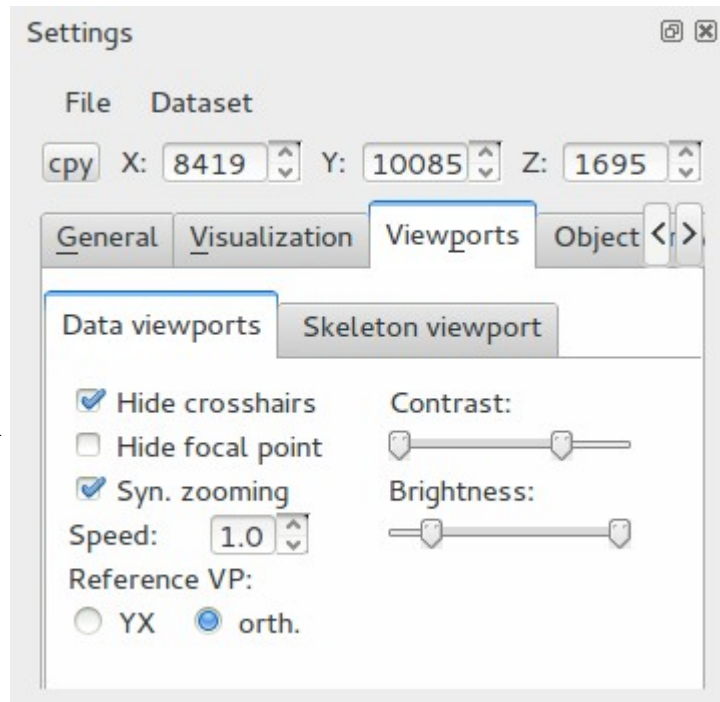
- Color scheme: Define which coloring scheme you want to use
 - None: Use the colors saved in the skeleton files
 - Auto: Automatically assign colors



Viewports tab

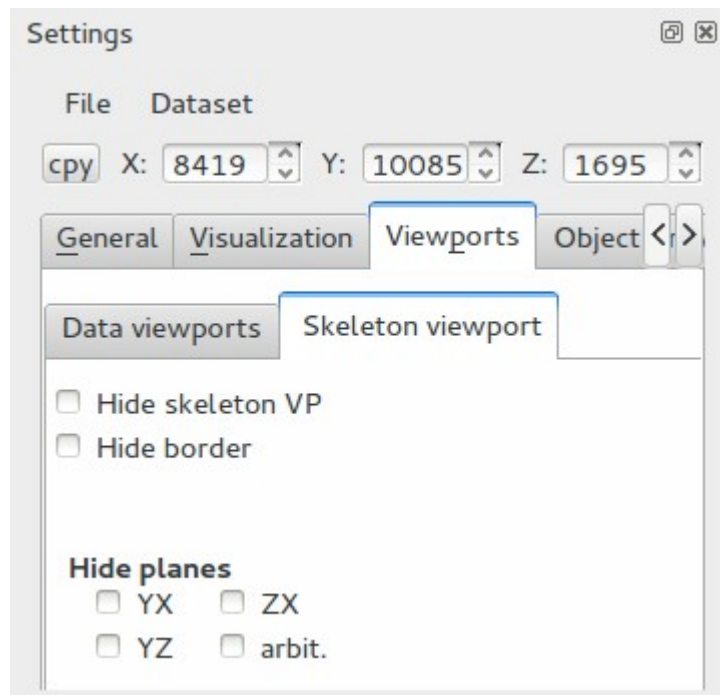
- **Data viewports:**

- Hide crosshairs: Hide/show crosshairs of the other viewports
- Hide focal point: Hide/show the red cross that marks the focal point/ current position of the viewports.
- Speed: Defines the movement/navigation speed.
- Contrast: Adjust contrast by thresholding pixel intensity from bottom and top.
 - Keep SHIFT + right mouse button pressed and move the cursor...
 - ... up/down for changing the upper threshold
 - ... left/right for changing the lower threshold
 - SHIFT + ...
 - UP/DOWN arrow keys for changing the upper threshold
 - LEFT/RIGHT arrow keys for changing the lower threshold
- Brightness: Adjust brightness
- Reference VP:
 - YX: The top-right and bottom-left viewports are orthogonal to the YX plane, i.e. they correspond to the YZ and the ZX planes, respectively.
 - orth.: The top-right and bottom-left viewports are orthogonal to the bottom-right viewport which is usually perpendicular to the tracing direction



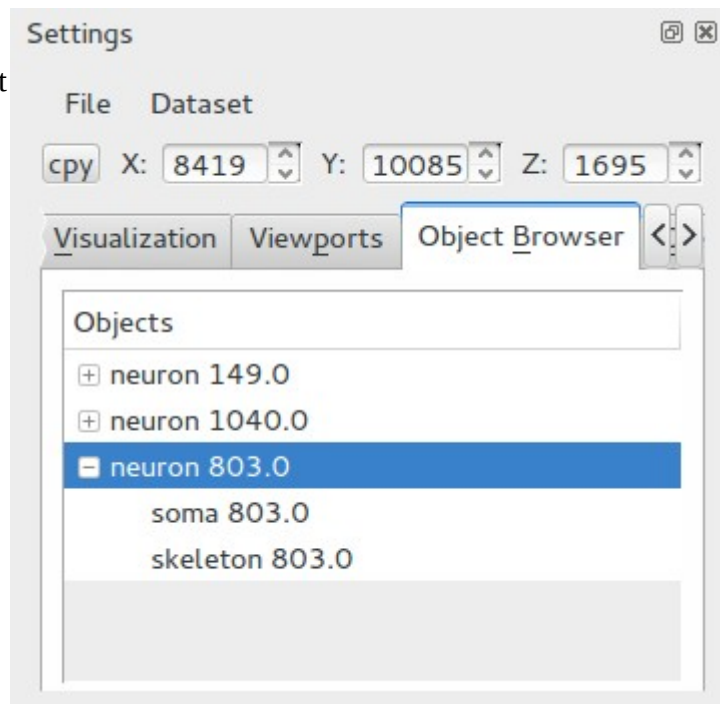
- **Skeleton viewport:**

- Hide skeleton VP: Hide/ disable the skeleton/3D viewport for faster rendering of the other viewports.
- Hide border: Hide the border of the viewport planes in the 3D viewport
- Hide planes: Hide the viewport planes/reslices in the 3D viewport and only show the border of the planes.



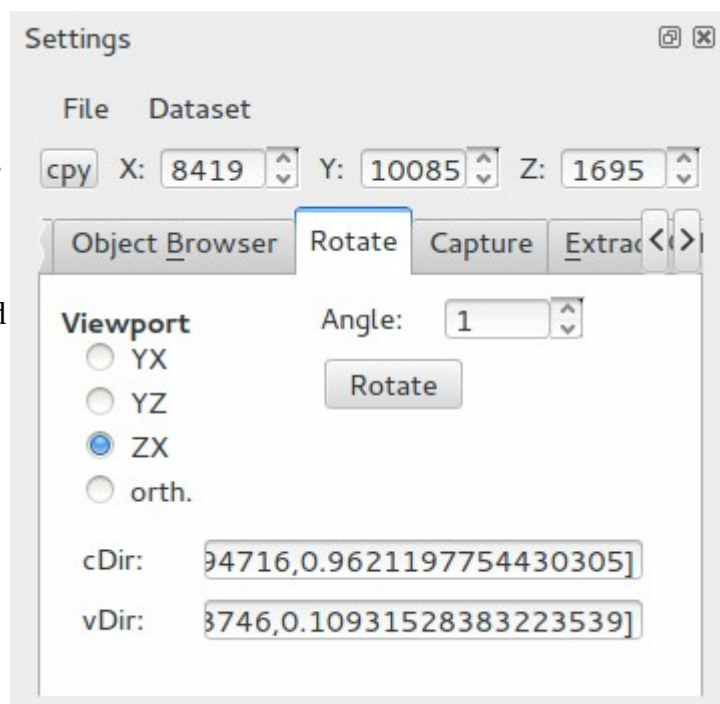
Object Browser tab (experimental)

- Object browser that shows the object types and IDs hierarchically.



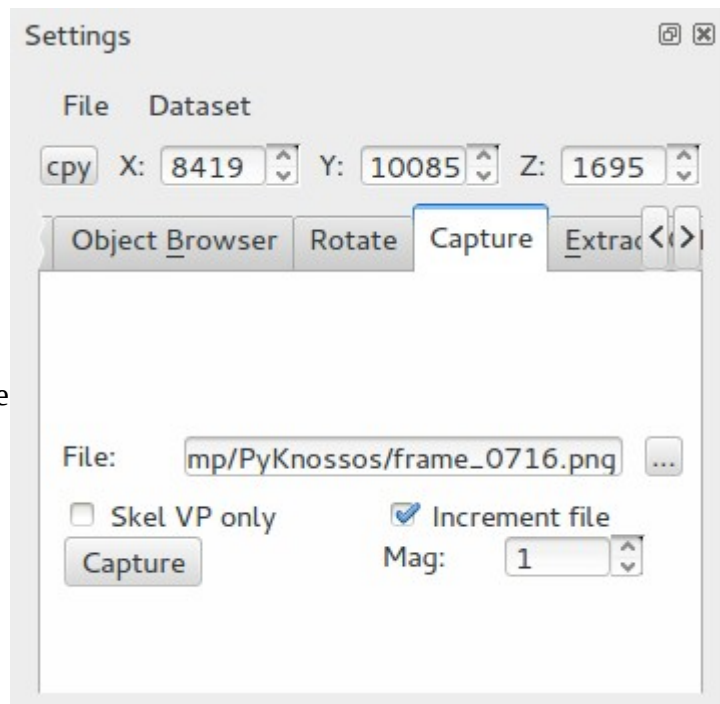
Rotate tab (experimental)

- Allows to rotate the bottom-right viewport by a defined angle around one of the 4 viewport axes (YX, YZ, ZX, orth)
- Alternatively a camera direction/ plane normal (cDir) and camera view up vector (vDir) can be defined for the bottom-right viewport.



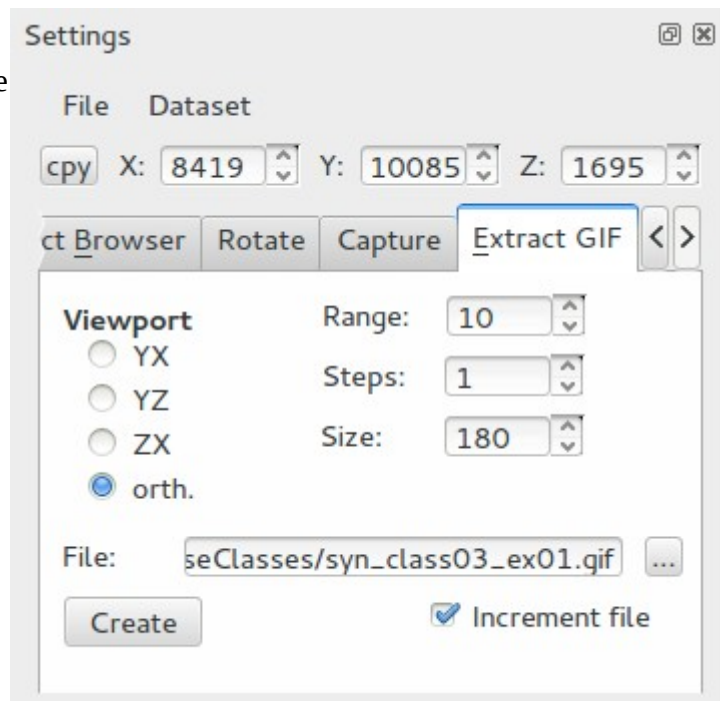
Capture tab (experimental)

- Capture/ C key: Captures an image of the currently visible viewports and saves it as a png-image at arbitrary resolution.
- File: Base file path and file name
- Increment file: Increments the file number automatically whenever an image is captured
- Mag: Magnification/resolution of the captured image.
Note, on Windows OS higher magnifications might results in run-time errors.



Extract GIF tab (experimental)

- Captures a series of images from one of the viewports (YX, YZ, ZX, orth) and saves it as an animated GIF-image.
- Range: Number of steps up and down from the current position
- Steps: Number of slices per step
- Size: Size of the extracted image around the current position (in pixels)
- File: Base file path and file name
- Increment file: Increments the file number automatically whenever an image is captured

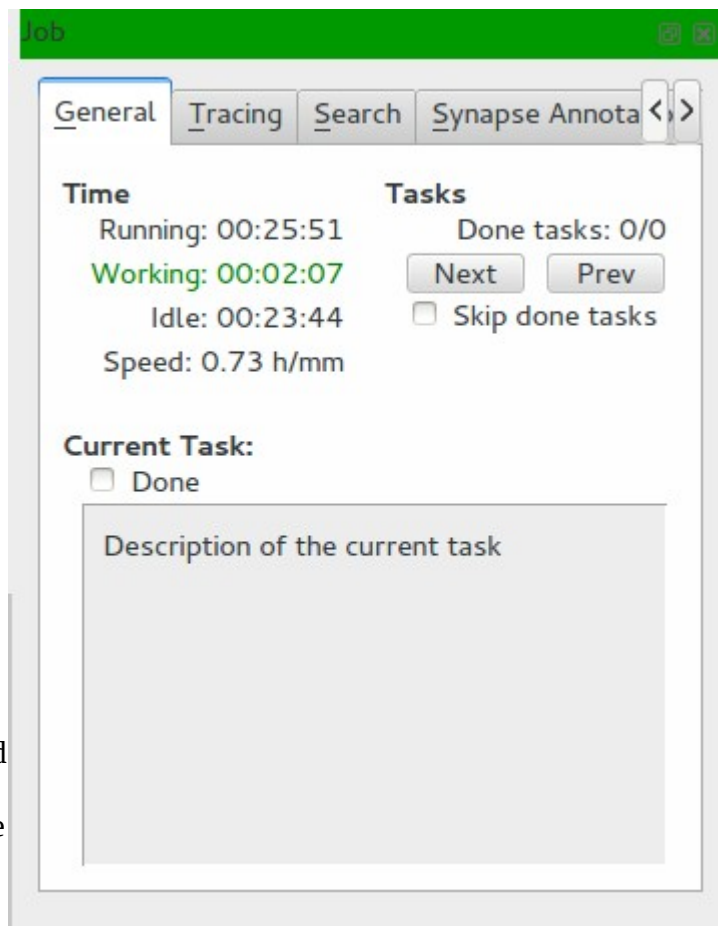


The *Job* window

- Double-clicking on the window title maximizes the menu windows and hides the skeleton viewport.
- The color of the *Job* window title bar changes with the working speed (from green: good working pace to red: too slow working pace)

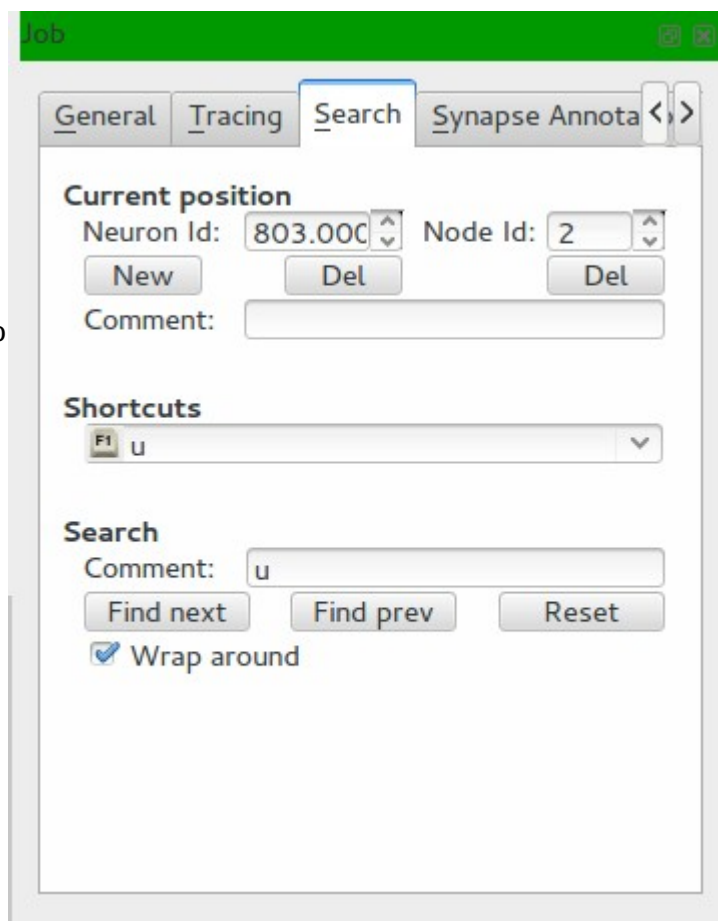
General tab

- **Time:** Shows the run time of PyKnossos, the actual working time and the idle time, as well as the working speed.
- **Tasks:** Some jobs might consist of multiple independent tasks (e.g. tracing or synapse annotation). Here you see the total number of tasks and the number of remaining tasks for the current job. Allows to browse the task list.
- **Current task:** Description of the current task (e.g. tracing or synapse annotation).



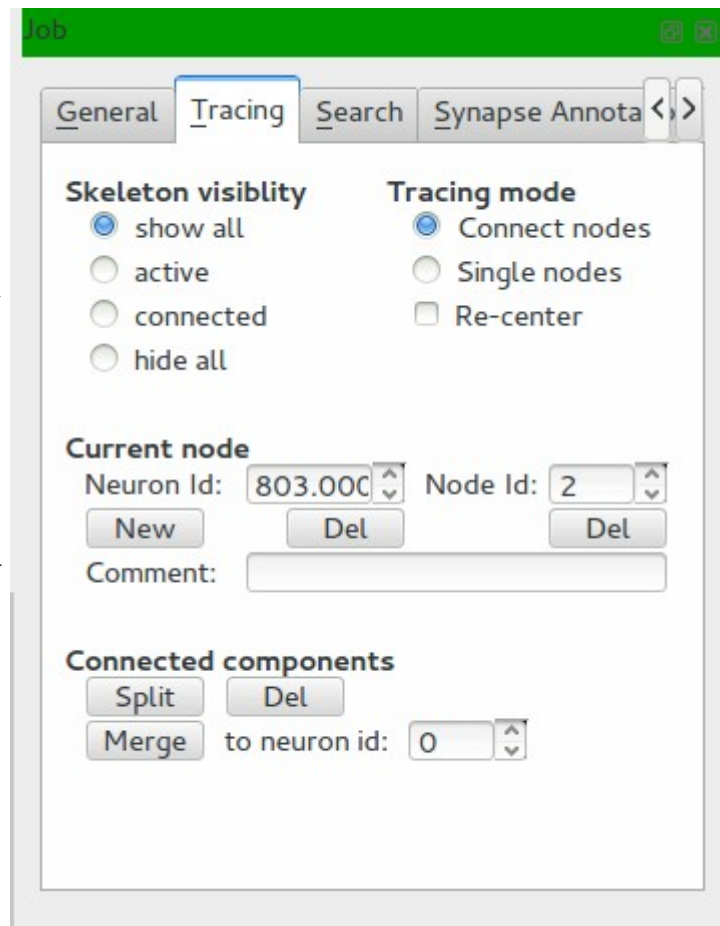
Search tab

- **Current Position:**
 - Shows/sets the neuron id and node id of the currently selected object
 - New: Add new neuron
 - Del: Delete current neuron/ node
 - Comment: Add a tag/comment to the current node.
- **Shortcuts:** Assign comments to the shortcut keys F1-F5
- **Search:** Search for tags/comments.
 - Find next: Search/ jump to the next object with that tag (**shortcut N**)
 - Find prev: Search/ jump to the previous object with that tag (**shortcut P**)
 - Reset: Reset search mask
 - Wrap around: Search through all objects starting from the current object.



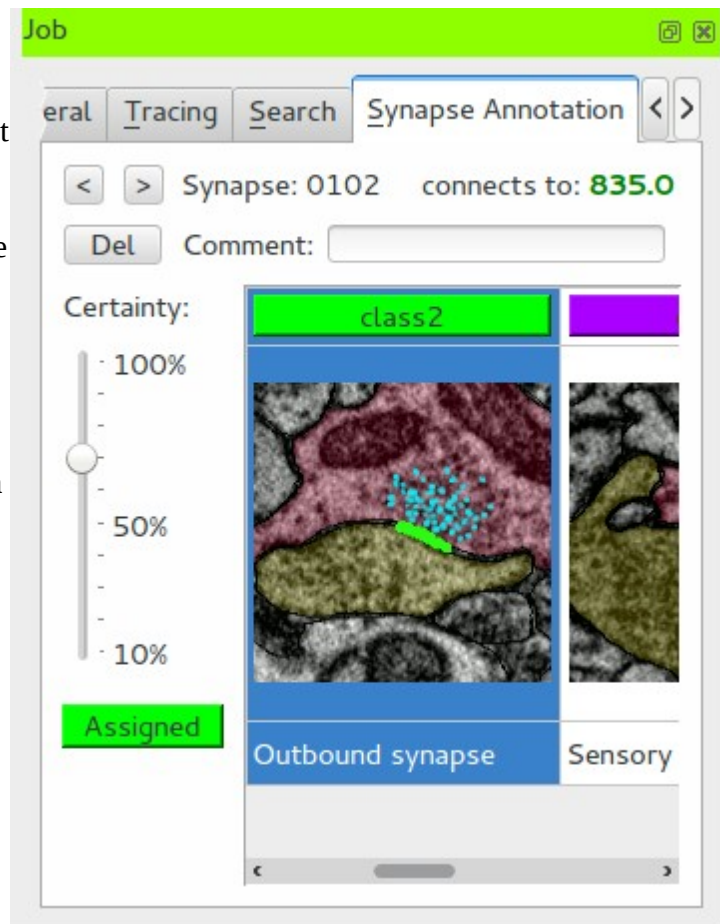
Tracing tab

- **Skeleton visibility:**
 - show all neurons
 - show active neuron only
 - show connected components: Show only nodes that are connected to the current selected node
 - hide all skeletons
 - **Shortcut H:** Switch between show/hide all
- **Tracing mode:**
 - Connected nodes: connect a newly put node to the previously selected one.
 - Single nodes: Put single, unconnected nodes.
 - Re-center: Re-center data viewports at newly put node
- **Current node:** (see *Search* tab)
 - Shows/sets the neuron id and node id of the currently (selected) object
 - New: Add new neuron
 - Del: Delete current neuron (left button) or current node (right button).
 - Comment: Add a tag/comment to the current node.
- **Connected components:**
 - Connected components: A set of nodes that are connected by edges
 - Note, often it is necessary to first split connected components by deleting an intermediate node or deleting an edge between two nodes (see "Delete/add an edge" below)
 - Split the connected components of the currently selected node to a new neuron
 - Delete the connected components of the currently selected node
 - Merge the connected components of the currently selected node with neuron id ...
- General tracing instructions:
 - Make sure you are in **Tracing Mode** (see *General* tab in the *Settings* window).
 - **Placing nodes:** Click with the right mouse button into one of the data viewports
 - Depending on the tracing mode (see *Tracing* tab in the *Job* window) the new node will be connected to the previous one or not.
 - Active node: The currently selected node
 - **Re-center data viewports at active node:** Press S key
 - **Delete active node:** DEL key
 - **Move/drag node:** Keep SHIFT key pressed and left click on a node to move/drag it.
 - **Delete/add an edge:** To add/ delete an edge between the active node and another node, click on the other node with the scroll-wheel while CTRL key is pressed.
- For additional tracing information see the chapter "Tracing Instructions" below.



Synapse Annotation tab

- Certainty: How certain are you about your synapse annotation?
Shortcut keys: 1: 10% to 0: 100%
- Comment: Add a tag/comment to the current node.
- Assign(ed) button: Assign the current certainty level and synapse class to the current synapse
- Next/Prev: Browse through the synapses in the order they have been put.
- Del: Delete current synapse
- Synapse class browser:
 - Allows to browse/ select a custom set of synapse classes.
Shortcut keys: F1-F12
 - Each class can have a
 - custom description
 - custom color
 - multiple animated examples of the corresponding class
- Select the synapse class and use the scroll-wheel to browse the examples of each synapse class
- Double-click on the example image starts/stops the animation of the example



How to prepare your own datasets

- Like [KNOSSOS](#), PyKnossos dynamically loads cubed datasets.
 - The image stack has to be divided into cubes of 128x128x128 pixels
- To cube your image stacks you can use the Python script provided by the KNOSSOS team: https://github.com/knossos-project/knossos-python-tools/tree/master/knossos_cuber
 - Data type for the cubes: unsigned 8-bit integer
 - PyKnossos supports binary raw data cubes and png or jpg compression
- The cube files have to be stored in a particular folder structure:
 - `/mag*/x****/y****/z****/BaseName_mag*_x****_y****_z****BaseExt`
e.g. `~/myStack2/mag1/x0000/y0053/z0040/myStack2_mag1_x0000_y0053_z0040.raw`
- Storing multiple sets of cubes at different resolutions allows for seamless zooming:
 - `/mag1/`: Contains the cubes with highest resolution
 - `/mag2/`: First downsampled version of the stack (e.g. downsampling factor: 2)
 - `/mag3/`: Second downsampled version of the stack (e.g. downsampling factor 4)
 - etc.
- Note, the downsampling factors do not have to be isotropic for all dimensions
- Create a configuration file for PyKnossos as follows:
 - Create a new text file “YourDataset.config” in the parent folder of the cubed dataset (e.g. `~/myStack2/myStack2.conf`) and add the following lines:

```
[Dataset 1]
_BaseName = myStack2
_Description = "high quality"
_BaseExt = .raw
_DataScale = 10,10,25, 20,20,25, 40,40,40, 80,80,80, 160,160,160,
_NumberofCubes = 100,100,50, 50,50,50, 25,25,32, 13,13,16, 7,7,8,
_FileType = 0
```

- If you have multiple cubed versions/ channels of the same dataset, just append the configuration information for those to the config file, e.g.:

```
[Dataset 2]
_BaseName = myStack2
_Description = "low quality"
_BaseExt = .jpg
_DataScale = 10,10,25, 20,20,25, 40,40,40, 80,80,80, 160,160,160,
_NumberofCubes = 100,100,50, 50,50,50, 25,25,32, 13,13,16, 7,7,8,
_FileType = 3
```

- `_Description`: This short description string will be shown in the window title of PyKnossos when the dataset is loaded.
- `_DataScale`: Contains a list of the actual resolution in nm of the different magnifications of the datasets:
`x_mag1,y_mag1,z_mag1, x_mag2,y_mag_2,z_mag2, x_mag3,y_mag3,z_mag3, etc.`

- `_NumberOfCubes`: Contains a list of the number of cubes in each dimension of the different magnifications of the dataset:
`NcubesX_mag1, NcubesY_mag1, NcubesZ_mag1,`
`NcubesX_mag2, NcubesY_mag2, NcubesZ_mag2,`
`NcubesX_mag3, NcubesY_mag3, NcubesZ_mag3,` etc.
- `_FileType`: PyKnossos supports the following file types:
 - File type 0: binary image cubes (default Knossos cube format)
 - File type 1: png image cubes, slices arranged from left to right
 - File type 2: png image cubes, slices arranged in rows
 - File type 3: jpg image cubes, slices arranged in rows
- PyKnossos supports streaming the data cubes from local or online servers. Just append the following parameters to each dataset in your configuration file. If no user name and password is required, leave these parameters empty:

```
_BaseURL = http://yourserver.org/pathtothedata
_UserName = yourusername
_Password = yourpassword
_ServerFormat = 2
```

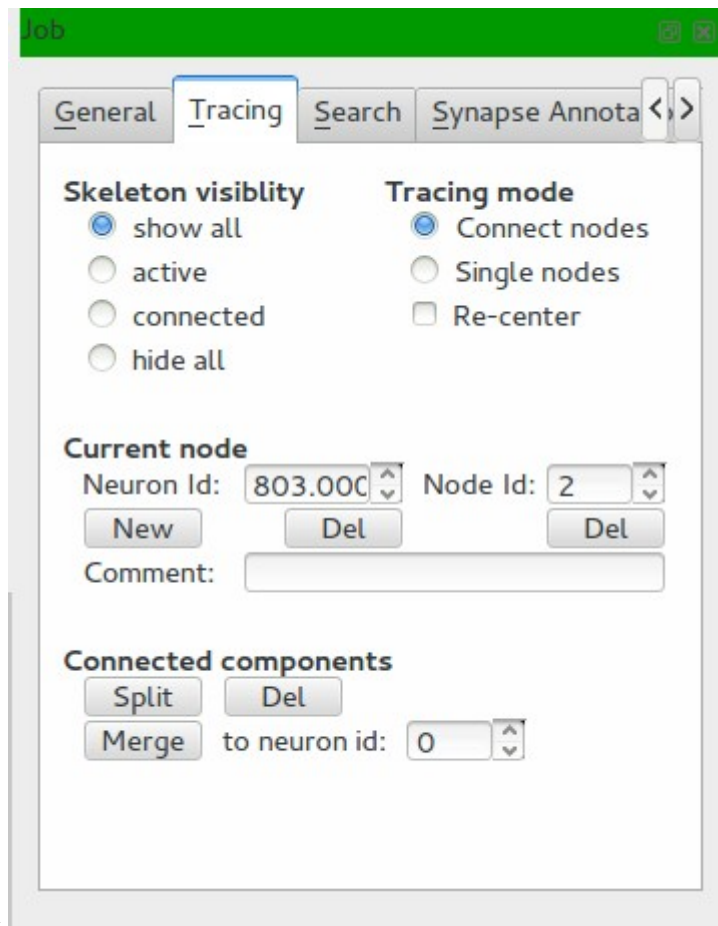
- PyKnossos supports various ways/formats of streaming the data from servers:
 - Server format 0: No streaming, the cubes are located on a local drive in the same folder as the configuration file.
 - Server format 1: The cubes are hosted by the openconnectome project (OCP/ neurodata.io)


```
_ServerFormat = 1
_BaseURL = http://openconnectome.org/ocp/ca/WannerAA201605/SBEM1/jpeg
```
 - Server format 2: Default HTTP accessible server

Tracing Instructions

General instructions:

- Make sure you are in **Tracing mode** (*Settings window* → *General* tab).
- **Placing nodes:** Click with the right mouse button into one of the data viewports
 - Depending on the *tracing mode* (*Job window* → *Tracing* tab) the new node will be connected to the previous one or not.
- Active node: The currently selected node
- **Re-center data viewports at active node:** Press S key
- **Delete active node:** DEL key
- **Move/drag node:** Keep SHIFT key pressed and left click on a node to move/drag the node.
- **Delete/add an edge:** To add/ delete an edge between the active node and another node, click on the other node with the scroll-wheel while CTRL key is pressed.
- Add comments/tags: Whenever possible, use the shortcuts (F1-F5) to assign tags/comments to nodes. You can also add tags/comments manually in the text fields in the *Search* or *Tracing* tab or use any of the shortcuts.
- I key: new neuron



How to trace

- You get seed/ revision nodes as starting points for the tracing.
- The seed and revision nodes are tagged with the comment "u", which indicates that this node has not yet been traced ("untraced node").
- Starting from this seed points you follow the neurites through the dataset
- Make sure, that the auto-saving mode is enabled
- Place nodes about every 10-20 steps
- Whenever possible, use the **orthogonal viewport** for tracing. In that viewport, the neurite cross-section usually appears most roundish and branch points are easier to detect.
- Always place the nodes in the center of the cross-section
- Whenever possible, use the keyboard shortcuts for tracing: Use the F and D keys for moving forward and backward, and the F1-F5 keys for assigning comments. For example F1 can be used to assign "u" and F3 is used to assign "" (empty comment), i.e. delete any comment
- Whenever the neurite is branching, you mark the branch point by setting a node with the tag "u".
- To re-visit tagged nodes (e.g. untraced branches/nodes), you can use the "Search" function. Whenever possible, use the P key (previous tagged node) and the N key (next tagged node) to quickly jump to untraced nodes.
- If you have doubts whether a neurite continues or not, tag it with the comment "?"
- Looking at the YX viewport is often useful at difficult locations, as it usually features the highest resolution.
- Monitor your tracing speed. The color of the title bar of the *Job* window changes its color according to your tracing speed.
 - **Green** indicates **good tracing pace**.
 - **Yellow** means that you are **rather slow**.
 - **Red** indicates that you are way **too slow** and you have to work faster and more focused or make somewhat bigger steps between nodes.

How to identify branch points:

- **Trace focused and carefully:** It is important that you do not miss any branch points.
- Branch points are sometimes difficult to identify. Sometimes it is useful to move up and down in the viewports in order to find a branch point.
- Hints for branch points:
 - Neurites often become thicker at branch points.
 - Neurites from different directions are fusing together.
 - If you find a sudden change of neurite direction this might indicate that you have just missed a branch point. Go back and find it!
 - It is much easier to find branch points, if you trace the neurite in the *orthogonal viewport* because there neurite has the smallest diameter and appears most roundish.

How to trace/check branch points efficiently

1. Tag the node as "untraced" branch point ("u")
2. Follow one of the branches until you reach the end of the process
3. Use the "Find (p)revious" function to search for nodes with comments "u"
4. Check if you have traced all the branches
 - **Use the S key to jump back to the currently selected branch point**
 - For example:
 - Scroll up in the orthogonal viewport, hit the S key, scroll down in the orthogonal viewport, hit the S key
 - Scroll up in the XY viewport, hit the S key, scroll down in the XY viewport, hit the S key
5. Remove the comment "u" if all branches at this branch point have been traced

Solving issues

1. **Problem: PyKnossos is reacting slow. Visualization is slow.**

Solution: There are several possibilities to improve the performance of PyKnossos (ordered by priority/ speed gain).

1. Turn off your anti-virus software. It might be scanning every data cube that is loaded into PyKnossos.
2. *Settings window → Visualization tab → Skeleton → lines/points*
3. *Settings window → Viewports tab → Skeleton viewport → Hide skeleton VP*
4. *Settings window → Visualization tab → Skeleton → Hide skeleton nodes*