Brief Manual for Proofreading Tool for the Output of the Automatic Segmentation Algorithm

Introduction

In this section we assume that the reader is familiar with the process of Serial Section Electron Microscopy and has a basic understanding of the nature of the data represented by the stacks of Electron Micrographs (EM). If not, the reader is advised to first look at http://www.synapse-web.org/ and specifically at http://synapse-web.org/learn/filo3D/howto.stm.

Basic Concepts and Some Terminology

In the following we introduce some basic concepts and terminology essential for understanding the Proofreading Tool (PRT).

PRT is MATLAB software designed specifically for rapid verification of the segmentations of stacks of Electron Micrographs produced by Automatic Segmentation Algorithm (ASA). In ASA the volume of the stack is viewed as composed from a number of *elementary blocks* of various sizes and shapes. These blocks are such that any neuronal process inside the stack can be represented as a simple collection of the blocks. A segmentation of the stack's volume, thus, is a partition of this set of elementary blocks into classes relevant to different 3D objects. One useful way to think of this particular approach is to picture a stack of flat jigsaw puzzles (or tilling puzzles) in which the pieces are assembled to represent particular neuronal protrusions.

Currently, the elementary blocks are built automatically based on definition that elementary block is a region within single EM section that can be said under "worst-case-scenario" to be surrounded on all sides by a lipid membrane. The process of breaking the stack volume into elementary blocks we call ***partitioning***. It is expected that ASA finds all (or nearly all) locations where a membrane could be in principle placed so that reshaping of the elementary blocks will almost never be necessary. Automatic congregation of such blocks into 3D objects follows partitioning and is performed based on criteria stemming from continuity in z-direction. The process of association of elementary blocks with final 3D objects we call ***assembly*** (also can be called clustering). A collection of elementary blocks (as assembled into single 3D object) relevant to a 3D neurite we call ***representation*** of that neurite.

With regard to the above, we introduce the following terminology:

- ***Segment*** is used here interchangeably with "elementary block", and refers to area in one EM section that is surrounded on all sides by features that can be thought of as lipid membrane.
- ***Object*** refers to collection of elementary blocks, both in the same EM section and across a series of EM sections, which are identified with one 3D object.
- ***Label*** is integer number enumerating 3D object in the segmentation. There are two kinds of labels: ***forward*** and ***final***. Forward labels are intermediate labels, e.g. those produced by ASA. Final labels are labels used to enumerate final objects, e.g. after proofreading. Objects with different forward labels may be assigned the same final label, but converse is not true.
- By ***p-map*** we understand a set of relations between forward labels and final labels – such relation specify what final label corresponds to an object with some forward label and may be written as a table "forward label → final label". For example:

map	Forward	1	2	3
	Final	3	3	3

- We suggest using following convention for identification of directions in the stack: up and down refer to two directions perpendicular to plane of EM sections and north-west-south-east refer to directions within single EM section.

ASA Segmentation and Proofreading

At current time ASA is prone to significant amount of errors. It utilizes, essentially, analysis of local image features under most conservative assumptions and is unable to account for higher-level reasoning often necessary to resolve ambiguous situations. Proofreading of ASA output is necessary to verify and either confirm or modify ASA decisions. At this time proofreading of ASA output is necessary in the entire volume of the segmented EM stacks.

Given that any proofreading that does not explicitly inspects 100% of segmentation volume is not guaranteed to be lossless, different strategies may be involved to minimize the chance of loss of information vs. the amount of manual labor involved. While better proofreading strategies is still an open topic, we advise to use so called ***all-major*** strategy for proofreading current ASA outputs, in which the operator is offered to inspect all neuronal processes, in the order of decreasing sizes, and is asked to finalize and confirm every one of them.

In all-major strategy, to reduce the amount of labor involved in proofreading, ASA generates a set of "significant" 3D objects, which it believes are parts of valid neuronal processes, as opposed to many small "noisy" objects, which may not be associated with any actual process. The ASA's intent is to create a list such that every actual neurite in the volume of the EM stack has at least one of its 3D fragments in the list. Operator is lead to examine all items in the set of "significant" objects, which is a subset of all 3D objects produced automatically during segmentation, and verify them.

The progress of proofreading is recorded in the form of ***proof-notes***. Proof-notes are electronic records attached to particular 3D objects that carry operator's confirmation (or rejection) that the neurite is correct, and a verbal note. Only those objects that have a proof-note indicating that operator examined and confirmed them are added to the final proofed version of the segmentation, all other ASA-generated objects are removed from the final proof.

A second-stage algorithm is also employed that allows extraction of smooth contours for neuronal processes based on object's partial representations created for the final proof. Thus, operator is not asked to create a "perfect" image of a neurite during proofreading, but rather only to select a "substantial" backbone of the segments for particular neurite.

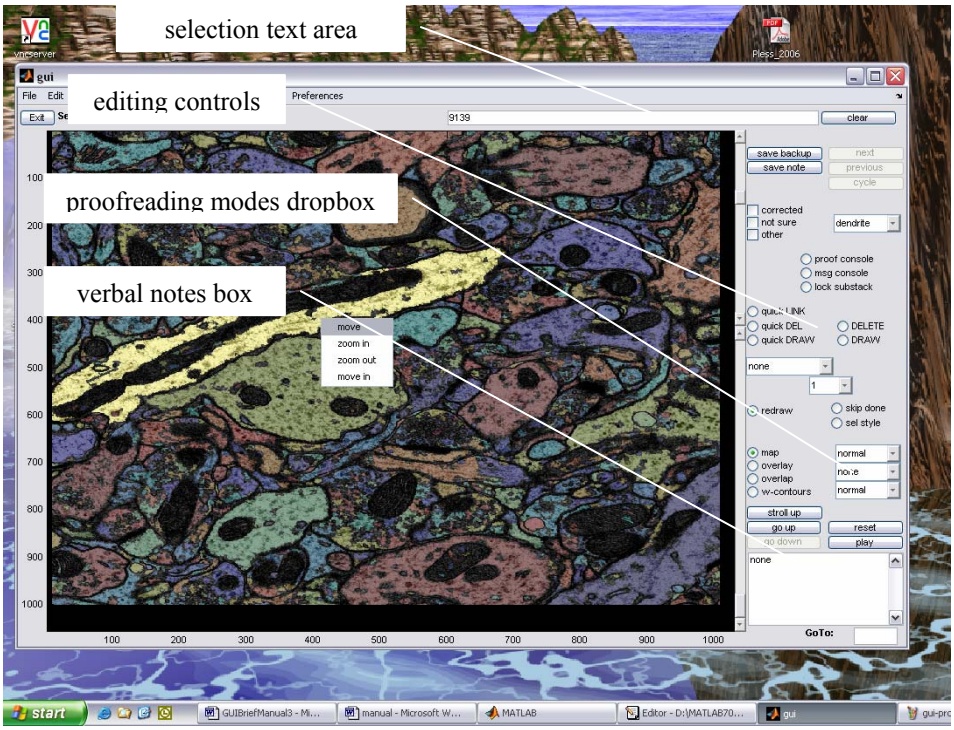
Currently, with all-major strategy it is expected that an operator proficient with PRT will be able to verify 3-5 cubic microns of neuropil per hour (approximately one 10μmx10μm section per hour). For example, a volume of 50 cubic microns from Rat neocortex an experienced operator should be able to finalize in 10-20 hours of work. Such volume typically would contain about 500 different fragments of neuronal processes (including dendrites, axons, spines and over-fragmented glia), from which 1-10 fragments may be still lost after proofreading.

The throughput limitation of all-major strategy is set by the time necessary to inspect "entirely-correct" segmentation and is about 20-30 cubic microns per hour.

Disclaimer

At this point, ASA + PRT package is essentially a ***semi-manual system*** aimed at reducing time and manual labor necessary for full reconstruction of 3D volumes of neuropil. As such, one should have reasonable expectations from this software. While we find that amount of time needed for full reconstruction of neuronal processes within 3D volume is significantly reduced, essentially almost all neuronal processes will require some degree of attention. The advantages provided by our system are that of reducing labor involved in drawing neuronal processes, tracking through locations which are unambiguously resolvable and providing hints for the behavior of neuronal process in locations which are ambiguous as well as providing guidelines for systematic reconstruction of the entire neuropil volume. On the other side, since for reliable 3D reconstruction practically all neuronal processes will have to be inspected, using our software may be a minor advantage over simple stamp-tracking method. With further improvements of image processing and analysis this situation may change; however, we are hopeful that proofreading tool will provide capacities general enough to be applicable also with such future ASA versions.

Quick Start Reference Guide



Flip	Browse down	Browse up	Quick LINK
q	w	e	r
Quick DELETE	Quick DRAW	DRAW	DELETE
a	s	d	f
UNDO	Re-center	Cycle	Redraw
ctrl-z	x	c	l

- **Quick overview of processing sequence** – when we receive stack of EM images, we first register the images and then use scripts written for MATLAB to automatically segment the stack. The segmentation is generated in chunks of about 10 sections with two upper/lower sections overlapping with the adjacent chunks: e.g. stack of 30 sections would be segmented as the collection of following sequences 1-11, 10-21, 20-30.
  Each of the subseries is an independent block of data and is proofread independently: operator examines the subseries in computer guided process and marks corrected neurites as "correct" for the final proof. Once all true neurites had been marked in subseries, the operator repeats process for the other subseries. Once all subseries in the stack had been proofread, they are combined back in a continuous sequence using the overlapping parts of the stacks by a MATLAB script.
- **Installing PRT tool** – PRT tool is a MATLAB script and as such requires MATLAB on the computer. To install PRT, simply copy m- and fig-files from the PRT GUI package to ***work*** subdirectory in your MATLAB installation directory.
- **Starting PRT** – to start PRT GUI, launch MATLAB and navigate in MATLAB to the directory containing ASA output data files. Then type "startgui" in MATLAB command prompt. In "startgui" dialog select the data file for the subseries you will be working on, and click "Load".
- **Navigating in the stack** – to browse through the stack use "move up/down" buttons, or the hotkeys. If in the first/last section of the subseries, and the subseries is not locked, this may cause PRT to load the adjacent subseries – it is possible to browse in this way

through the entire stack, not just "local" 10 sections. To change the field-of-view, use the hotkeys or "move"-command from popup menu (right-click). To adjust zoom, use either the hotkeys or "zoom in/out" command from popup menu. You may also adjust zoom to extend to current selection using "move in" command from popup menu, or the relevant hot-key. You may select 3D objects by left-clicking and deselect by shift clicking (left or right). Selection may be also manually specified by typing the labels in the selection text area. Selection may be highlighted using selection mode dropbox right below the proofreading modes dropbox.

- **Checking registration** – before you start proofreading, briefly browse through the stack to see if there are significant problems with automatic registration at any locations. You may enable "overlay" mode for that, which will overlay three consecutive sections in RGB channels. If you notice large areas where registration is damaged, this will likely be the cause of numerous mergers of different neurites within that area. It may be worthwhile to send such subseries back with indication where the problems with alignment had occurred and request re-segmentation. We may be able to fine-tune registration in the problematic area and save a lot of proofreading time for you.
- **Saving work in PRT** – to save proofreading session click "save backup" button. Even though PRT maintains backup as you go, ***always manually save backup*** before exiting or leaving the workplace for long time. Saved proofread session may be resumed from the same point you left at later time.
- **Resuming work in PRT** – to resume previously saved proofreading session follow the steps for starting PRT. If a backup file for the subseries data file you selected is found, PRT GUI will display prompt asking whether it should load the saved session.
- **Starting proofreading** – to initiate proofreading, select "major" from proofreading modes dropbox, and bring up proof-console using "proof-console" switch. We also advise to lock the current subseries using "lock substack" switch.
- **Selecting items from proof-list** – to select an item from proof-console simply click on it. You may also advance through the proof-list systematically using "cycle" button, which will advance to next not-attended object in the proof-list. Proof-list is sorted in the order of decreasing 3D volumes. For each item in the proof-list, you need to inspect the neurite that contains it in every sections of the subseries, and either confirm or reject it via the proof-note.
- **Confirming items from proof-list** – to mark a corrected neurite for the final proof and create a proof-note, mark one of the checkboxes "corrected", "not sure" or "other", and then click "save note" button. Proof-note also is saved whenever any of the checkboxes is marked and you select new item from proof-list either directly or using "cycle" button (in that case you don't need to explicitly click "save note"). Only the items that have a "corrected" proof-note will be present in the final proof, all other items will be automatically removed for the final proof as "noise".
- **Operating PRT's proof-notes system** – proof-notes saved with "corrected" flag will indicate that 3D objects are correct and should be added to final proof; proof-notes saved with both "corrected" and "not sure" flags indicate objects which requires further attention, but still should be added to the proof. A verbal note may be saved with such proof-notes using verbal notes box; such notes may be later selectively pulled up in proof-console using proof-console filters and the verbal note may be inspected. Proof-note saved with only "not sure" flag indicates that the object should be removed from the final proof. Regarding these behavior, the last proof-note overrides all previous proof-notes.
- **Editing segmentation** – in case a neurite's representation is not correct, PRT GUI allows alterations of the segmentation using "editing controls".
  Editing controls allow linking and drawing. Linking allows operator to merge two 3D objects into one by reassigning them to the same neurite. "quickLINK" operation, usually

enabled by default when new proof-list item is selected, indicates that subsequent clicks will link objects to the one identified by the proof-list. “quickLINK” should be used to lump together 3D pieces of neurite which was automatically oversegmented.
Drawing operations allow to explicitly “recolor” single segment in single section with new label. Drawing operations use the first label in the selection text area as the “color” of the pen. “quickDRAW” operation recolors a continuous shape with one click, while “DRAW” operation allows to draw new shapes using a square-shape pen of variable sizes. The size can be selected in the small dropbox with numbers near editing controls. Delete is a drawing operation which replaces labels with zeros, thus removing shapes from the segmentation. Drawing operations should be used to break merged neurites apart. Unlike linking, drawing only affects the current section, thus if two neurites are merged together, one of them should be redrawn in each section of the stack.
All editing operations may be rolled back using “ctrl-Z” key.

- **Forming substantial backbone -** drawing as well as linking of all small fragments is time consuming. This is why the segments marked by operator are considered as the backbone indicating ***approximately*** the shape of the neuronal process. The final contours for neurons are drawn using a special post-processing procedure based on a “watershed”. Without going into the details of this procedure, it will suffice to say that it is capable of filling holes inside objects as well as interpolating parts of the external membrane that are only partially defined. Thus, it is not necessary to make all tiny fragments belong to the target dendrite, but only as many as needed to obtain a “good enough” backbone for the dendrite’s profile.
  The question what is “good enough” is mostly intuitive and depends on the goal and experience. For the beginners we suggest following guidelines. A segment should be linked or a draw marker should be placed in the region if
  - this region is separated from already marked areas of the same neurite by dark features as pronounced as the nearby external membrane;
  - there would be left a gap in the neurite’s boundary 100-200 pixels long [500nm-1micron], or if such boundary is not sufficiently straight or simple, keep in mind that whatever is not selected will be dropped automatically from the final proof;
  - there would be a left blank area 100x100 pixels large or larger;
  - small, thin, dark or “noisy” protrusion need markers along their centerlines, if such thin protrusions are desired to remain in the final proof – draw them.

  When you place markers manually using draw operation, such markers do not need to follow the boundary precisely, it is generally sufficient if such marker are placed inward no more than ½-marker-width away from the true membrane. The shapes, which such markers touch, will be considered as parts of the same marker. Thus, it is extremely important to make sure that markers you place do not touch pieces of the nearby neurons. At the same time, few markers placed strategically to touch most of the pieces that do belong to the target object may suffice to fill its interior entirely. Please look at the examples provided below to get a better idea (Fig. 8, 9 and 16).
- This brief reference is not a substitute for the rest of the manual. Be sure to read through Tips&Tricks below, and at least briefly review full reference and quick start walkthrough sections.

Useful Tips and Tricks

This is a list of some recommendation concerning better practices for using PRT GUI. This is by no means a complete list and any additions are welcomed.

- **A useful procedure to follow when separating merged objects:**
  - First focus on one of the neuronal protrusions from those that have been merged; without paying attention to the rest of the merged objects make sure that this

neuronal protrusion is *fully contained* within the merger. Select type for that neuronal protrusion, make sure "corrected" check box is checked and press "save note" to create proof-note *without loosing* current position in the proofing list.

- Press "clear", select any small fragment from nearby, just to use it as a label for the object you are going to decouple from the merger, and either quick draw or delete/draw over a portion of one of the other neuronal protrusions merged together: this will be your first recolor correction. *After your first correction* deselect any editing operations and simply click to select the rest of the merged objects. Then click-select the original merger. You do this first correction to initiate a proof-note for the neurite you are creating and, thus, identify it as the target for all further linking; having selection cleared and your object selected first will identify it as the color for all further drawing operations. Thus, even though you will have two objects selected, all quick edit operations will still proceed with the ID you started with. Having entire merger and current object selected together will help you to see the degree of corrections required to *accurately and completely* decouple the current neuronal protrusion from the merged bundle; you may miss some of the pieces that should be recolored otherwise
- Use quick DRAW and DRAW operations to "recolor" segments belonging to this neuronal process in all other sections of the stack. You may also use quick LINK operation on objects which are not part of the merger.
- Don't' forget to delete original small fragment if it is not actually a part of thus reconstructed neuronal process.
- Click "save note" when you are done with the current neurite and repeat above steps until all neurites are made separate.

- **If you are not sure whether two 3D objects belong to one neuronal process or not**, **leave them *disconnected*** and create a "not sure" proof-note to identify the nature and location of your confusion. In uncertain cases it is better to leave things in pieces because *it is by far much easier to link two pieces that were by mistake left disconnected, than to decouple two objects that were by mistake merged.*
- **When fixing a merger, some times it is possible to simple quick DELETE wrong parts** and either drop them, if they are just on the edge of the field of view or otherwise are considered not important (e.g. – small), or redraw them later. *Never try to fix merger by deleting entirely one of the merged objects – this may result in complete loss of such deleted object.*
- **Do not stop tracing objects in the middle.** If you cannot understand what happens at some location and cannot find continuation of the neuronal process, be sure to create a "not sure" proof-note with note clearly identifying location and nature of the problem.
- **When proofreading the main goal is to make sure that the *object is not merged* with *something else*.** If you loose a valid part of some object, it is likely to show up later at some point in the proofing list and you will have chance to catch it later. There is no way to catch missed "merger" error other than re-inspecting the entire stack.
- **When you finished proofreading, click "save backup" button to make final backup file.** I also tend to create a *not-sure-only* proof-note with a note saying something like "STOP PROOFREADING, STANDARD STOPPING CRITERION". File ***gui.dataXXX.mat*** is the main result of your work, where XXX references the first section in the subseries. As is evident from the name, this file is subseries specific and will not be overwritten by PRT when you start working on the other subseries. Thus, there is nothing you need to do after creating this final backup file. However, to be on the safe side, I adopted the practice of maintaining a safe folder for finished proofs to where I copy files ***gui.dataXXX.mat, gui.buffXXX.mat*** and file ***gui.backup.mat*** renamed into ***gui.backupXXX.mat***. Note that ***gui.backup.mat***, which contains the copy of state of PRT

GUI, is not subseries specific and will be overwritten the first time you save backup when proofreading a different subseries.

- **When you need to recolor lots of smaller disjoint pieces, it is faster to place square markers that touch all these pieces.** You may then use one or two quick-draw operations to recolor the whole thing, although it is not even necessary since PRT will treat all pieces touching thus placed markers as parts of these markers anyway. At the same time, pay attention that none of your square markers touches a shape that belongs to a different nearby neuronal protrusion.
- **Linking implies that all segments that have been connected to selected object by ASA, or earlier linked to it by operator, will be automatically relabeled.** Some times this may result in merging different neuronal process in current or other section. If you see this happen, rollback as many links as necessary to get rid of the problematic merger, and then quick DRAW, quick DELETE, DRAW or DELETE to fix the problem.
- **Rarely membrane between two neuronal processes in some section will be lost by ASA**, in which case a single segment will result covering parts of two different neuronal protrusions. Quick-draw will not remove this error; use DELETE to entirely clear one of the parts of such segment or draw a "empty" membrane separating two pieces of such segment. You will need to use DRAW or quick DRAW operation afterwards to color relevant area with the label corresponding to proper neuronal process.
- **Some times mitochondria or terminal boutons are badly over-segmented or completely blanked out from ASA segmentation because ASA treated such region as possible membrane.** In this case it may be quicker to draw manually the outline of such objects rather than linking a host of tiny fragments. Use DRAW with sufficiently large pen size to place boxes approximately equidistantly inside the area covered by such tiny segments.
- **When reconstructing an axonal bouton, you will only need to link fragments along its boundary.** If automatically selected fragments follow the boundary well, this is generally good enough even if there is a number of pieces left unconnected inside. This is because of the watershed post-processing routine which will fill holes for you.
- **The issue of "sufficiently good" backbone representation is a difficult topic mostly because it is so much intuitive.** And it is really important topic because this is one of the major factors that affect proofreading time. I learned this on intuitive level and it is hard for me to explain the distinction. Some introductory remarks have been provided in the quick start section. One other criterion that may be helpful is this: if you imagine that the fragment in question is absent from the segmentation, then if the "hole" in the "host" neuronal process is smaller than about 50-100 pixel in diameter, such fragment may be skipped. The "goodness" of backbone, in the end, is determined by the result of watershed post-processing, thus you may learn to estimate the goodness by trials and errors after examining "watershed" contours on your proofread data. Generally, you should aim at the smallest amount of corrections that still produce backbone satisfactory for final contours.
- **When you browse through the last portion of the proofing list (generally last half or two third), most of the items there will need to be dismissed because they will be rather tiny fragments of already finalized objects.** That is, "corrected" checkbox should be unchecked for them. It is convenient at that point to go to Preference Editor and select default state for "corrected" as "unchecked". It is convenient to have default state for "corrected" as "checked" when you just begin going through the proofing list since all of the things that you see will be new neurites.
- **If you link a segment to the object which had been previously finalized, GUI will print in bold a message in the notification area that the link target is an object that had been previously inspected.** This notification will stay up at least for about 0.25s. This will be helpful for many minor fragments for which it is unclear at first glance

whether they are parts of already finalized objects or not. Thus, in later portion of the proofing list, if you are ready and you see such message, you may immediately continue on to the next item because it means that the object in question had been attended before, and thus already had been finalized.

Also, generally when you trace some object and at some point link it to a "previously inspected" object, there is no reason to trace this object further, since everything beyond the point where you linked to "previously inspected" object should have been verified by now.

- **During proofreading of our datasets we discovered that there is rapidly decreasing chance to find a new "significant" neuronal process in the last half of the proofing list.** Generally, an *early stopping criterion* may be applicable to stop proofreading before reaching the end of the full proofing list. The early stopping criterion that I use is this: size of the largest object left in proofing list (shown after letter "M:" in proof console title bar) is below or around 500 pixel for 4nm/pxl resolution, and at least two consecutive pages of 50 items (or one page of 100 items, including those that belong to earlier finalized objects) from the proofing list had been skipped without finding a "significant" new neuronal process. "Significant" here is determined by the proofreader; one suggestion is to understand by this "axon or dendrite, or large fragment of glia, not immediately on the edge of the stack".
- **Generally, pieces of almost all neuronal processes in the volume can make it into some portion of the proofing list.** This rule however has exceptions: some thinner axons may be entirely filled with a mitochondrion along their entire fragment within the stack, or some axonal terminal boutons may be extremely densely populated with vesicles. In these cases such parts of neuronal protrusions may fail to score even one "significant" object into the list prepared by ASA. There is no way to catch such objects other than by looking for large holes in the final proof. We estimate that there may be 1-20 instances of such objects per 1000.
- **When tracing neurite into first/last two sections, keep in mind that these will be used to bind segmentations in adjacent subseries.** For this reason try to be extra-careful in those sections and if you see something that you feel may be differently interpreted by proofreader doing the adjacent subseries, better delete such thing entirely. Remember that mistakes in the two border-sections have potential to damage work on the entire adjacent subseries.

Proofreading in miniseries

For the purpose of working with larger datasets with our proofreading software on workstations with limited memory size, proofreading of larger EM stacks is organized as a sequence of subseries of sections from the full stack, which we call miniseries or ministacks. Miniseries are typically 12-22 sections long, they are organized in a way such that the lower two sections of one miniseries overlap with the upper two sections of preceding miniseries, and so on. When proofreading miniseries, operator should proofread each miniseries independently. After all miniseries have been finalized, they may be connected based on their overlapping parts and all miniseries may be combined into one continuous piece.

Thus, we advise to always proofread one miniseries at a time and to not try to finalize "one neuronal process" at a time over all miniseries. However, if the purpose of the proofreading is not full volume segmentation but rather a selected subset of neuronal processes from it, it is possible to trace particular objects over all miniseries in the stack.

PRT GUI allows navigation across miniseries boundaries for "on-site" resolution of ambiguous places near the bottommost or uppermost sections of the miniseries. When you click "go up" or "go down" while in the uppermost or bottommost sections, PRT will automatically load the adjacent miniseries and display it. PRT will attempt to carry over current selection into

the adjacent miniseries. However, operator is not advised do make any corrections while inspecting EM in the miniseries other than that which is been proofread. Moreover, such functions as changing item in the proofing list or saving proof-notes will be disabled while operator remains in the "wrong" miniseries. Once an ambiguous place had been resolved, always return to the correct miniseries to make any changes and create/save proof-note.

Reading data files for the adjacent miniseries is a time consuming operation, thus we do not advise to cross over the miniseries border unless it is really required to understand a situation in the currently proofread miniseries. To avoid accidentally crossing into and loading adjacent miniseries, always select "***lock substack***" when beginning proofreading. If PRT GUI finds enough physical memory, it will attempt to maintain the copy of the last loaded adjacent subseries in the buffer. Although this will speed-up the process of reloading this subseries, it still may take some time.

Appendix A: Quick Start Example Walkthrough

This section presents a quick start example walkthrough for using PRT. The quick start example is given for a sample ASA data file ***example.estack001-006.mat*** provided with this package.

PRT is a script written for MATLAB, thus you should launch MATLAB before using PRT. To run properly in MATLAB, files from PRT package should be placed into subdirectory "work" of MATLAB installation directory. For example, if MATLAB is installed in C:\MATLAB, the files should be placed into directory C:\MATLAB\WORK.

Once MATLAB is running and assuming that PRT files are in the correct directory, navigate to the directory containing ASA data files using MATLAB disk navigation tools. For this example, you should navigate to the directory containing ***example.estack001-006.mat*** file supplied with PRT package (this may be your MATLAB "work" directory). When you are working with other data, this may be a separate project folder where you keep ASA data files.

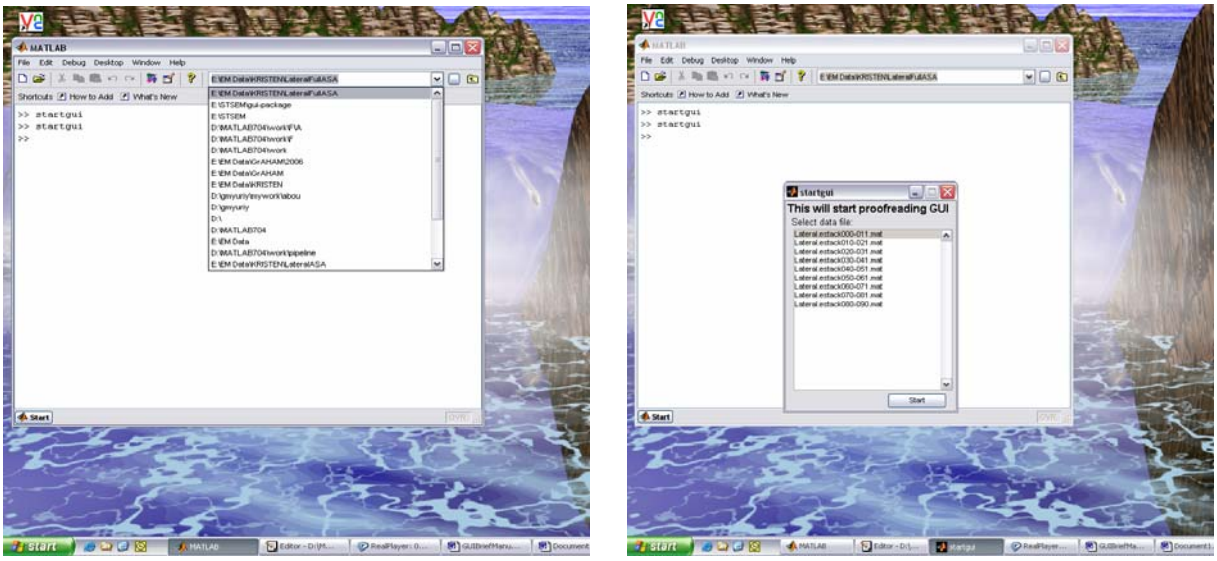


Figure 1 Navigate to project folder and execute *startgui*

To start PRT in MATLAB, type startgui in command line and press <enter>. This should bring up data-selection dialog. This dialog will list MATLAB data files in the current directory (it will hide support files used by PRT (LINK HERE)). We name ASA output files are named in the following convention "PREFIX.estackAAA-BBB.mat", where PREFIX refers to the project name, AAA refers to the number of the first section in the series and BBB refers to the number of the last section in this series. In this case you should see "example.estack001-006.mat", click-select this data-file in the window and click "start" button. This will load the data to the memory and start PRT's initialization. If a backup file from previous work session is found, a pop-up window will appear asking whether you want to load it. Answer "No" and wait until PRT finishes initialization.

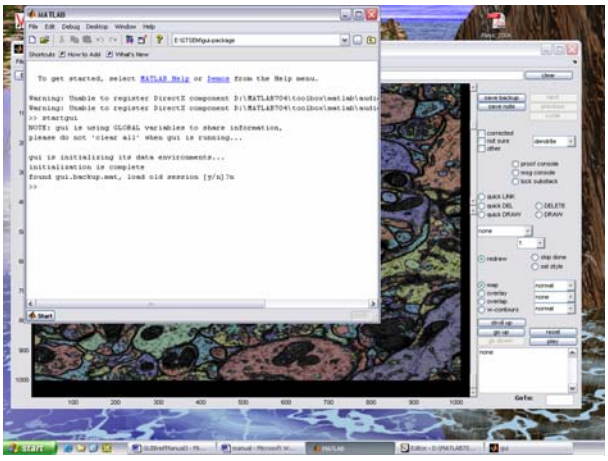


Figure 2 PRT may ask you to load backup file

Once PRT has initialized, GUI will appear with the first section of miniseries shown. By default, GUI displays overlay of color-coded segmentation and original EM images. Color-coded segmentation means that one object in segmentation is shown with one color. For this particular example, there are about 150 final neuronal processes and about 8000 distinct objects generated by ASA, so that some colors will not be very well distinguishable by eye, but they ARE different. You may adjust level of mixing between color-coded segmentation and original EM by using "mixer slider" to the right of the main graphical canvas. You may browse through the sections using "up/down" buttons or "section position slider". Note that the number of the section you are in is displayed in upper-left corner of GUI. The asterisk next to the section number indicates that you are in the bottommost or uppermost section of the series.

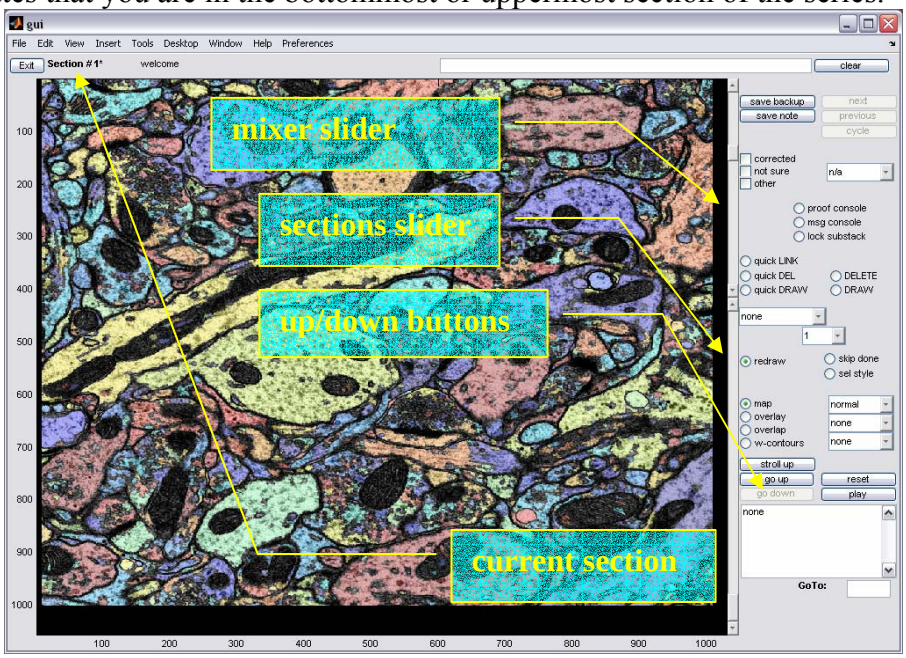


Figure 3 Navigating around EM stack

Take a moment to familiarize yourself with PRT interface. You may select objects by clicking on them in the main window. When you select on an object, its numerical label will appear in "selection area" as well as a message reading "Selected OBJECT-ID" will appear in "notification area". Note that when you click an object a 3D object gets selected, thus, if you browse to another section, selection will follow the profile of this 3D object. Each click adds to the current selection, you may de-select objects using SHIFT-CLICK. Also, you may define selection by entering coma-separated list of numerical labels of the objects you want to see into the selection area. You can have GUI highlight selected objects via selecting "normal" or "cluster" selection styles from "selection style" dropbox. Right clicking in the main window will also bring up pop-up menu which allows to zoom in/out around position of the click, re-center field of view around position of the click ["move"] and re-zoom around current selection ["move in"].

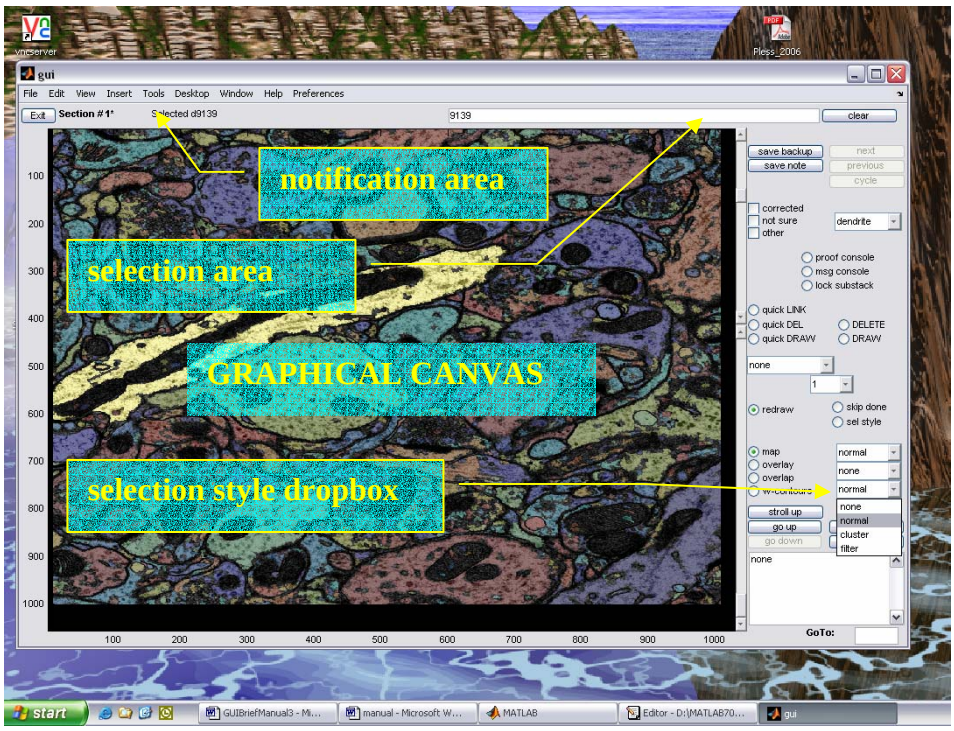


Figure 4 Selecting objects in segmentation

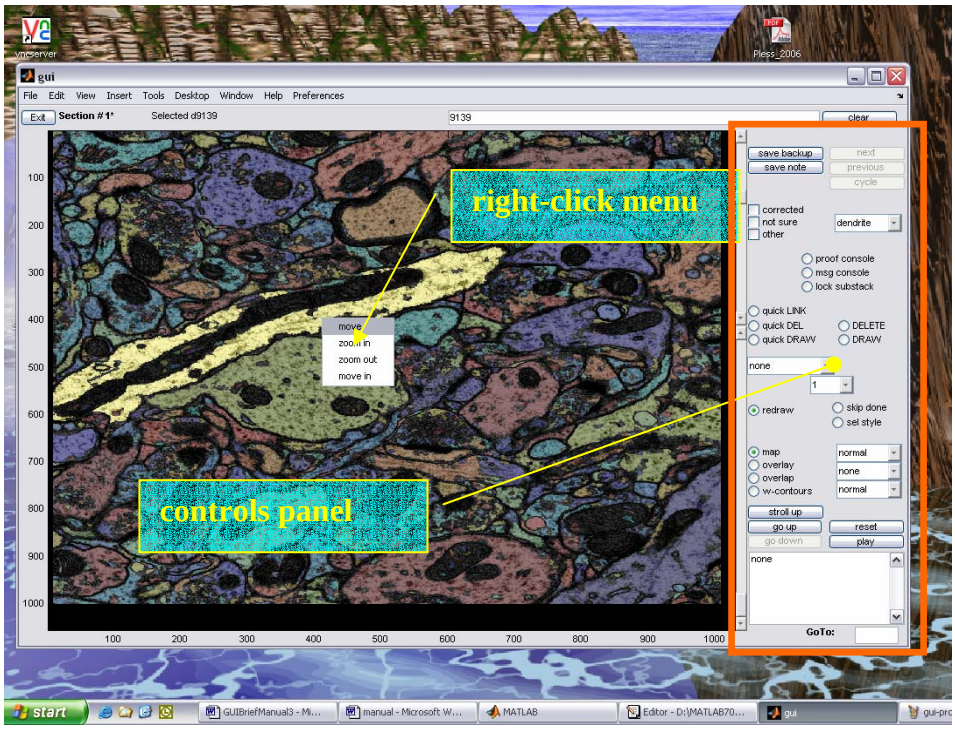


Figure 5 Right click calls pop-up navigation menu

To begin proofreading session, click on the "proofreading modes" dropbox and select "major". This will initialize proofreading mode for all-major strategy. There are two important auxiliary windows associated with proofreading session: proof console and messages console. Both can be brought up by clicking on corresponding radiobuttons in the middle of the controls panel, so please activate them now (Fig.6). You should see two additional windows appear as shown below.

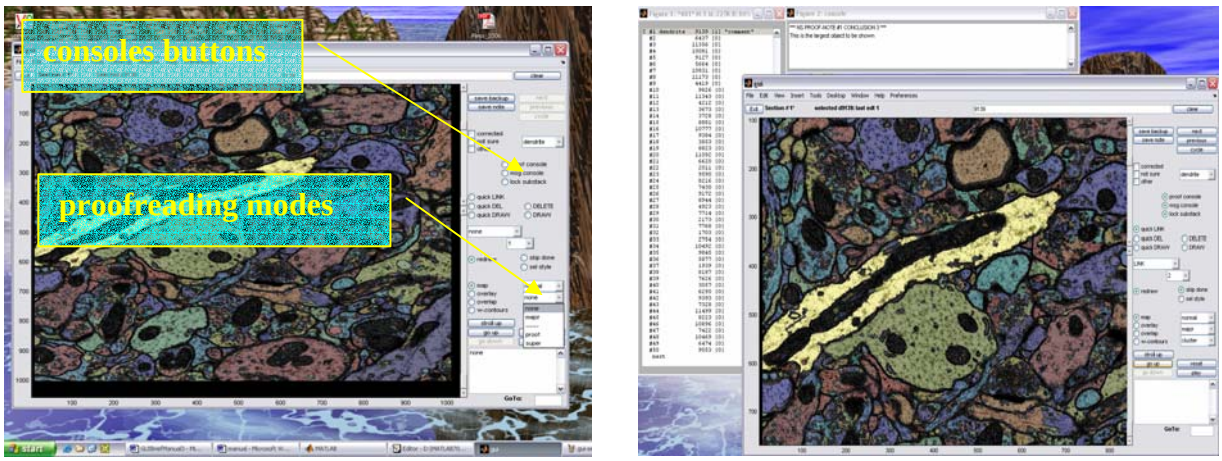


Figure 6 Initialize proofreading mode by selecting *major* from proofreading modes dropbox and activating proof console and messages console

Proof console is used to display and interact with the list of significant objects, as determined by ASA. The format of the items in the proofing list is as follows: position of the item in the list, object's primary type (such as dendrite, axon or glia, if any), object's label, whether the object had been ever examined before (letter "C" in front of the record indicates that the object has record of being inspected before), whether the object had been attended in current proofreading session (flag "1" in square brackets indicates that the object has being inspected during current proofreading session) and whether the object has any additional comments. In the title bar of the proof console PRT will print total number of items in the list, total number of neurites so far confirmed by the operator, largest size of the object left in the list below the current one (pixels), and estimated percentage of the total volume taken by the neuronal processes confirmed by the operator. Messages console is used to communicate to the operator the content of the comments, if any, left previously for selected item in the proofing list.

Items in the proofing list are normally ordered with respect to their volume. However, in this example the items had been manually ordered for illustration purposes. You may select item from proofing list by click-selecting it in the proof console or using "cycle" button. We suggest using "cycle" button to systematically advance through the list. To start proofreading, click on "cycle".

For each item in the proofing list operator should identify neuronal process containing shown segment and make sure that representation of this process is correct. By this we mean that, most importantly, there are no segments that belong to other neuronal processes and are attached to it, and secondly that all sufficiently large segments that do belong to it are labeled with the final label of this neuronal process. Operator should use editing operations if corrections to the segmentation are required to fulfill the above requirements. Finally, operator should create and save a proof-note describing the status of the neuronal process.

In the following we will present example of few items from the sample data stack illustrating these operations.

- The first item in the list is object #9139. This is a dendrite, it has some smaller fragments (e.g. just above the mitochondrion around position [300,400]) that are disconnected as well as some large blank area inside the dendrite corresponding to mitochondrion. Pieces that are disconnected can be attached using "linking" operations. To link smaller fragments mentioned above, select "quick LINK" and click on these fragments. As you do so, they will change color to yellow and become highlighted.
  Blank areas may be filled using "drawing" operations. To fix the blank area inside the dendrite, select "draw" and click along the centerline of the mitochondrion to create a band of yellow squares running near the center of the blank area. Make sure these squares do not extend outside of the dendrite.

Drawing as well as linking of all small fragments is quite time consuming. This is why the segments marked by operator are considered as the backbone indicating ***approximately*** the shape of the neuronal process. The final contours for neurons are drawn using a special post-processing procedure based on a "watershed". Without going into the details of this procedure, it will suffice to say that it is capable of filling holes inside objects as well as interpolating parts of the external membrane that are only partially defined. Thus, it is not necessary to make all tiny fragments belong to the target dendrite, but only as many as needed to obtain a "good enough" backbone for the dendrite's profile.

The question what is "good enough" is mostly intuitive and depends on the goal and experience. For the beginners we suggest following guidelines. A segment should be linked or a draw marker should be placed in the region if

- this region is separated from already marked areas of the same process by dark features as pronounced as the nearby external membranes;
- there would be a gap in the neuronal process boundary otherwise 100-200 pixels long [500nm-1micron], or if such boundary is not sufficiently straight or simple;
- there would be a blank area otherwise 100x100 pixels large or larger;
- small, thin, dark or "noisy" protrusion need markers along their centerlines, if such thin protrusions are desired to remain in final proof.

When you place markers manually using draw operation, such markers do not need to follow the boundary precisely, it is generally sufficient if such marker are placed inward no more than ½-width away from the true membrane. The segments, which such markers touch, will be considered as parts of the same marker. Thus, it is extremely important to make sure that markers you place do not touch pieces of the nearby neurons. At the same time, few markers placed strategically to touch most of the pieces that do belong to the target object may suffice to fill its interior entirely. Please look at the examples provided below to get a better idea (Fig. 8, 9 and 16).

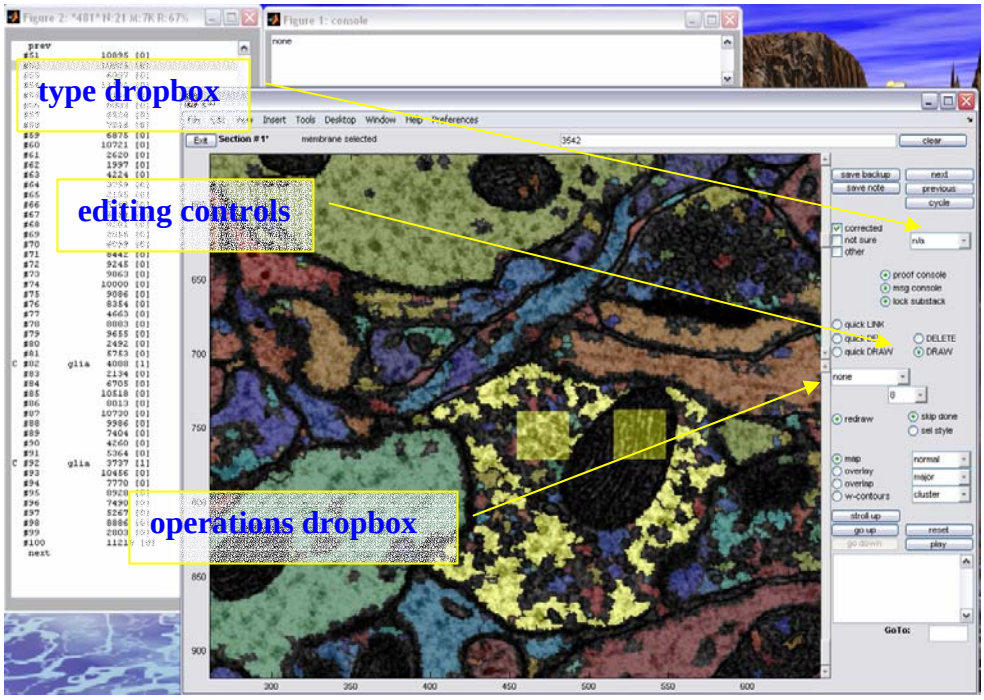


Figure 7 Editing controls and example of a "sufficiently well" represented axon

In specific case of d9139, it turns out that there is no need to link anything at all. If you, for a moment, imagine that smaller pieces on top of the mitochondrion were not there, you'd end up with one big hole inside the dendrite. This hole is fairly nicely surrounded by pieces that were "recognized" by ASA as parts of the same object and all "unmarked" areas have well pronounced membrane delineating them from the neighboring cells. Thus, in fact, there are no corrections needed at all for this dendrite. However, if you want to be on the safe side, you may

place couple of squares using “DRAW” along the portion of the mitochondrion that “contacts” the outer membrane, as shown in Fig. 8. Browse up and down from bottom to the top of the stack and make sure that similar situation holds everywhere. Place square markers similar to Fig. 8.

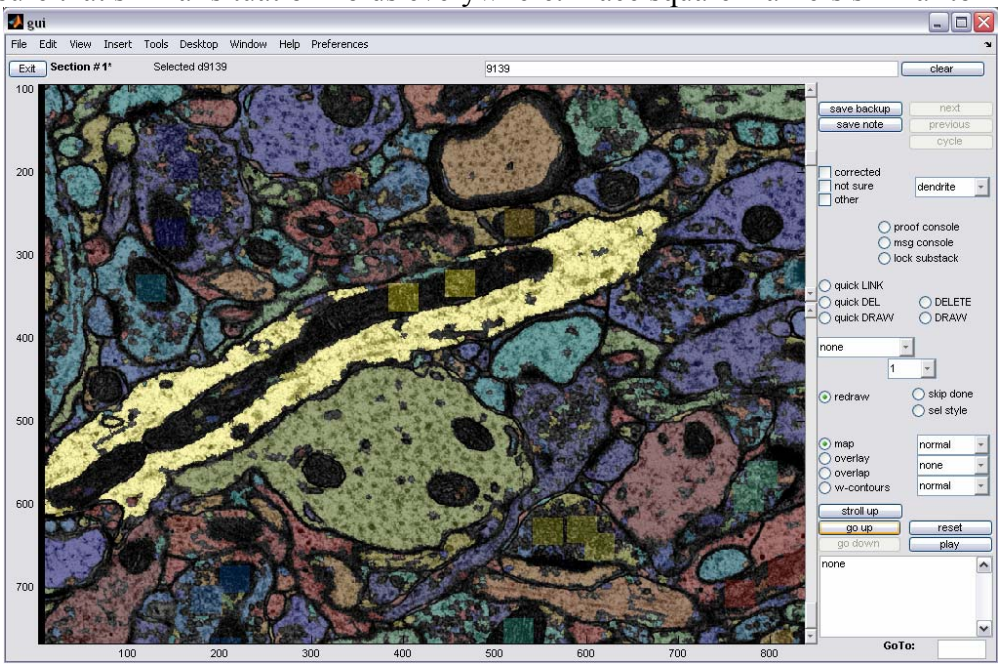


Figure 8 Corrections sufficient for d9139

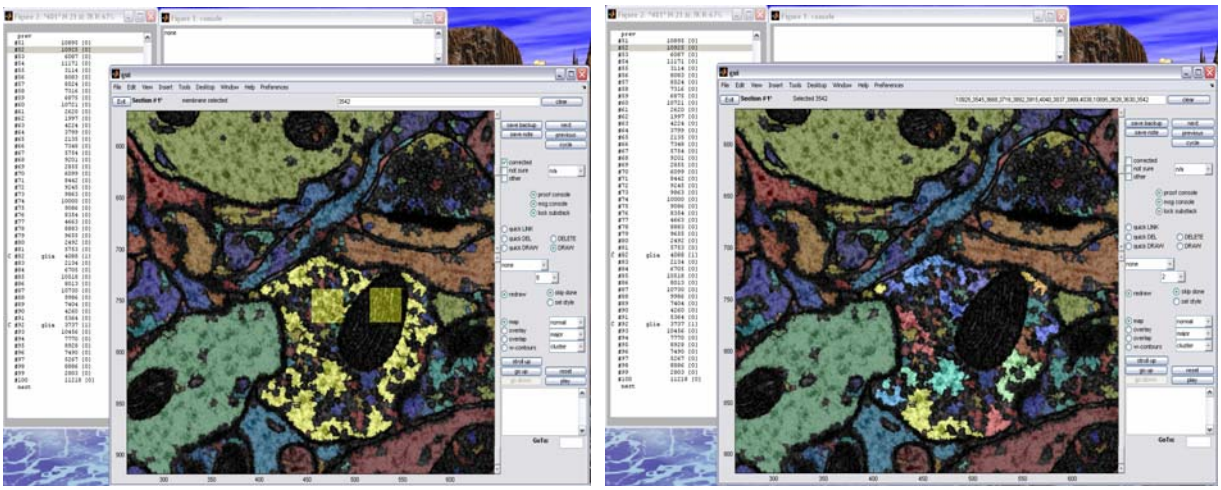


Figure 9 Example of severely oversegmented axonal bouton and how it may be fixed.

After you convinced yourself that d9139 is OK, you need to assign its primary type. Type selection is done using “type dropbox”. There are four primary types and six secondary types. Primary types include “dendrite”, “axon”, “glia” and “extra-cellular space”. Click on the types drop-box and select “dendrite” for this d9139. You will see that the message in the notification area changed to “Selected d9139”. The letter in front of the label will always be alpha-indicator of the selected object’s type.

Finally, you need to create a proof-note indicating that d9139 had been inspected, and is correct and should be added to the final proof. Proof-notes are defined via a set of “conclusion checkboxes”. To create a proof-note, you should check one of the “conclusion checkboxes” and click “save note” button.

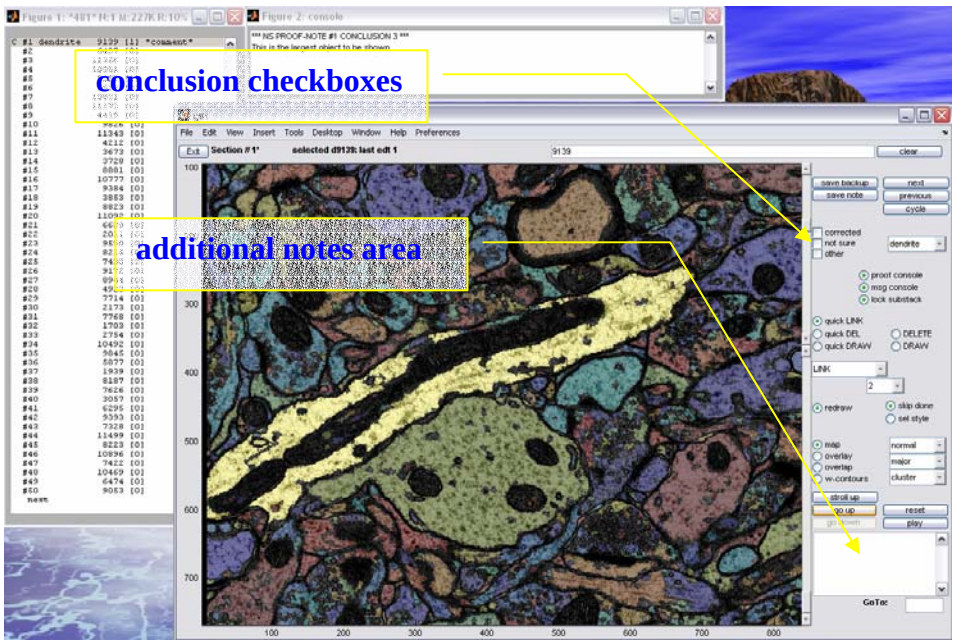


Figure 10 Create proof-note for finalized object.

"Corrected" checkbox means that you've inspected the object and it is correct, it should be added to the final proof and you don't need to look at it again (thus, "cycle" button will skip any parts of this same object that may appear later in the proofing list). "Corrected" checkbox also gets automatically checked every time you make a correction on currently selected object. Items that have no "corrected" flag set are excluded from the final proof (as if they were not there at all). "Not sure" checkbox means that there is something you are uncertain about. Such proof-notes are displayed in messages console when you select the item again in proofing list, they are used to communicate messages to yourself or a supervisor about locations which are not readily resolvable, or which appear to be interesting for whatever reason, and should be looked at again. Additional comments may be typed in the textbox in the lower-right corner of GUI. This comment should be descriptive enough for a different person to be able to locate in the stack the place that caused you to leave the comment.

For d9139 check "corrected" checkbox to indicate that it is all correct. To save proof-note, you may click "save note" button. Alternatively, you may just click "cycle" – every time selection in proofing list changes and there is a proof-note initialized, the proof-note is automatically saved.

- Next item #11356. Starting from the first section browse upward to get an idea of general behavior of this object. This object is generally well represented with the exception of a large blank area in its southern part adjacent to the image boundary. Because of the way the post-processing watershed is setup, a hole in object attached to the image boundary may remain as a hole. If that is undesirable, put one large square marker using DRAW operation in the center of the underrepresented area as shown in Fig. 11. In section 5 at the western edge of the object you may find a PSD with the nearby axon, use hot-key "q" to quickly alternate between original EM images and the segmentation to be able to see this PSD clearly. Thus, the object #11356 is a dendrite. Select "dendrite" for this object's type, then make sure that "Checked" checkbox is checked and click "cycle" to save the proof-note and advance to next item in the proofing list.

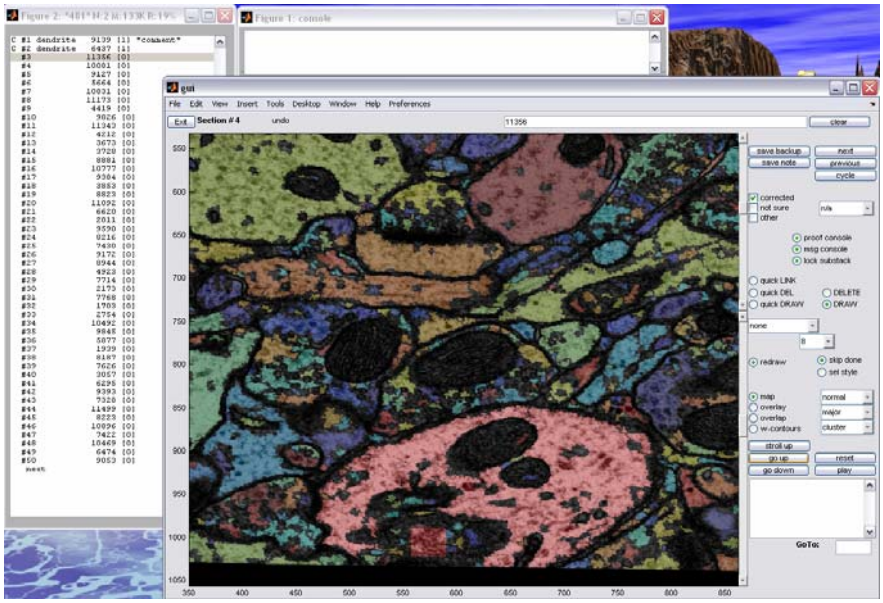


Figure 11 Fix southern hole in d11356 to insure proper watershed

- Next item is #9826. This is a large glial process with large mitochondrion in south. As before, browse once through the entire stack to get an idea of 3D look of this object. Use "q" hot-key to alternate between original EM and the segmentation for a clear view. You may connect thinner and brighter pieces to the left at the upper sections and around blue spine in the lower sections, however, for the mitochondrion it is more efficient simply to draw large square markers over it approximately through the centerline. See Fig.12 for example of corrections you should do. After you finished editing the segmentation, select "glia" for this object's type, make sure "corrected" is selected and click "cycle".

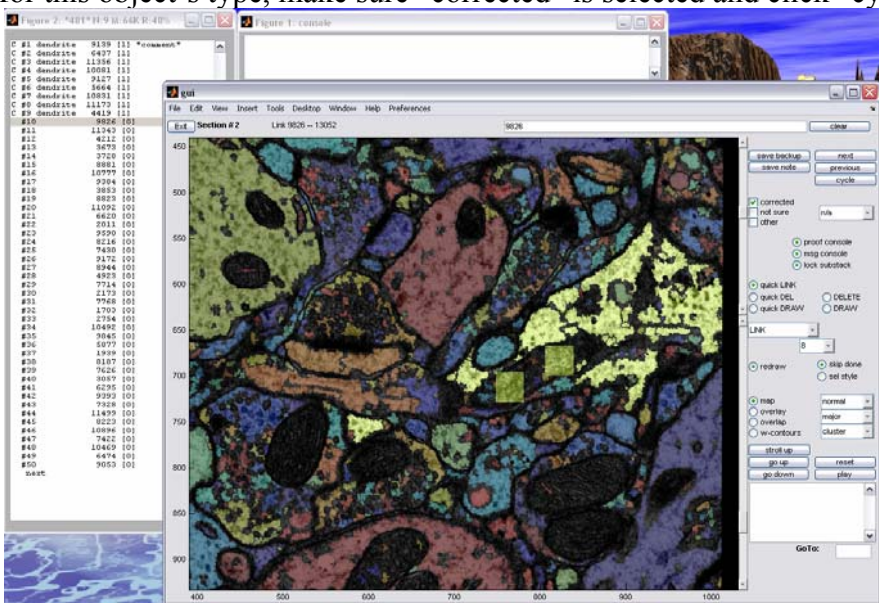


Figure 12 Restrict yourself to drawing large squares along centerline of mitochondrion in g9826

- Next item is #8881. This is axonal bouton filled with vesicles. Make a pass through the entire stack to see how this object behaves. Deal with "empty" axonal boutons as well as with severely over-segmented axonal boutons, i.e. where you cannot readily see any large elementary blocks, as with mitochondria – simply draw large squares approximately equidistantly over the central parts of them. Distance between squares should be approximately equal to the size of the squares. Once you are done, select "axon" as the type for #8881 and click "cycle".

- Most of the functions you used in proofreading are readily accessible from the keyboard. It is advised, in fact, to make use of the hotkeys whenever possible and avoid using buttons and dropboxes in the interface. The hotkeys layout is adjustable, however, we recommend following layout (also by default set with this example). Additionally, numeric keys "1-3" can be used to make neuronal process type assignment ("dendrite", "axon" and "glia" respectively).

Flip	Browse down	Browse up	Quick LINK
q	w	e	r
Quick DELETE	Quick DRAW	DRAW	DELETE
a	s	d	f
UNDO	Re-center	Cycle	Redraw
ctrl-z	x	c	l

- You may also configure GUI to make a specific selection for the state of some of the editing controls which will be set whenever you select a new item from the proofing list. From the main menu select "Preference Editor…": this will bring up preference editor dialog (Fig. 10). In the preference editor you may select the size of the default drawing pen ("dft pen size"), default state of the corrected checkbox "dft corrected state" and default state of the quick-link ("dft quick-link state"). You may also adjust the hotkeys via "hot-keys" dropbox. To do that, click on the dropbox and select the action hotkey for which you want to change. Once it had been selected a message in bold "press key associated with this function" will appear. Immediately after press the key you want to be assigned to this function. This will reassign the hotkey to selected function. You preferences selection is saved when you press "quit" button.

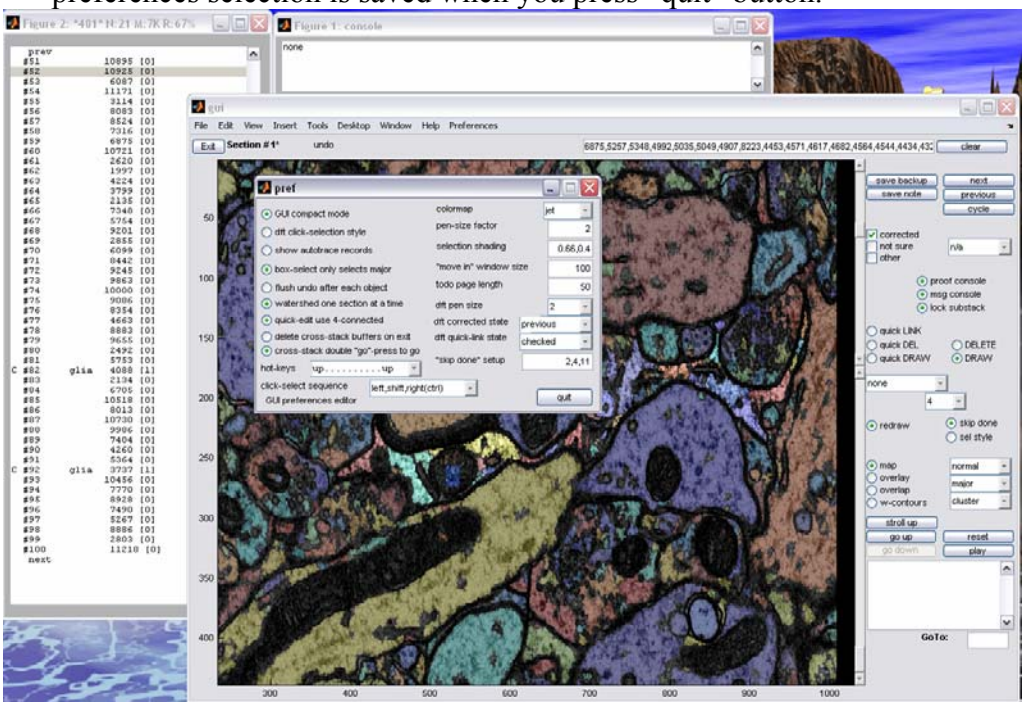


Figure 13 Preference Editor can be called from Preference submenu of main menu

- The next item in the list is #9384. This is an intricate fragment of a glial process, browse upward and downward few times and try to trace it fully. From this example try to choose for yourself the balance between the result and labor when dealing with glial cells. When finished
- Next item is #3853. This is a spine that extends above the last section in the stack. There is generally little or no editing that should be done with the spine, except for connecting them to proper dendritic shaft. #3853 is an outside spine, which connects to a dendritic shaft somewhere outside of the volume shown in the stack. After verifying that automatic

reconstruction of this spine is OK (by making a pass through the stack), make sure that "corrected" checkbox is checked and select " -spine" for this object's type, and press hotkey "c" to cycle to next item.

- Next item is #2011. This is a part of dendrite with spine extending to the south around sections 3-5, you just need to quick LINK the additional pieces. Press "1" to assign "dendrite" as the type, make sure "corrected" checkbox is checked and press "c" to cycle to the next item.

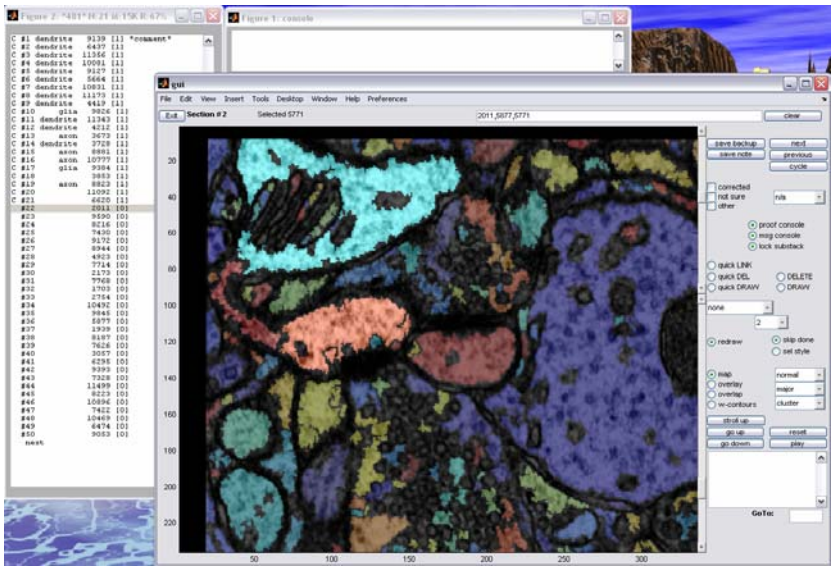


Figure 14 Quick LINK large spine below d2011

- #9590 is an axonal bouton that may look poorly segmented, however all corrections needed in that case are one box of size 8 approximately in the middle of the "empty" area in the axon's north in section 1 and smaller fragment in north-east in section 2. Otherwise, the outlines of this axon are sufficiently well drawn for post-processing to do a good job. Press "2" to assign "axon", make sure "corrected" is checked and press "c". P.S. in section 1 note small fragment of the same spine departing from north-east corner. You may link that piece as well [link 9590 – 4272].

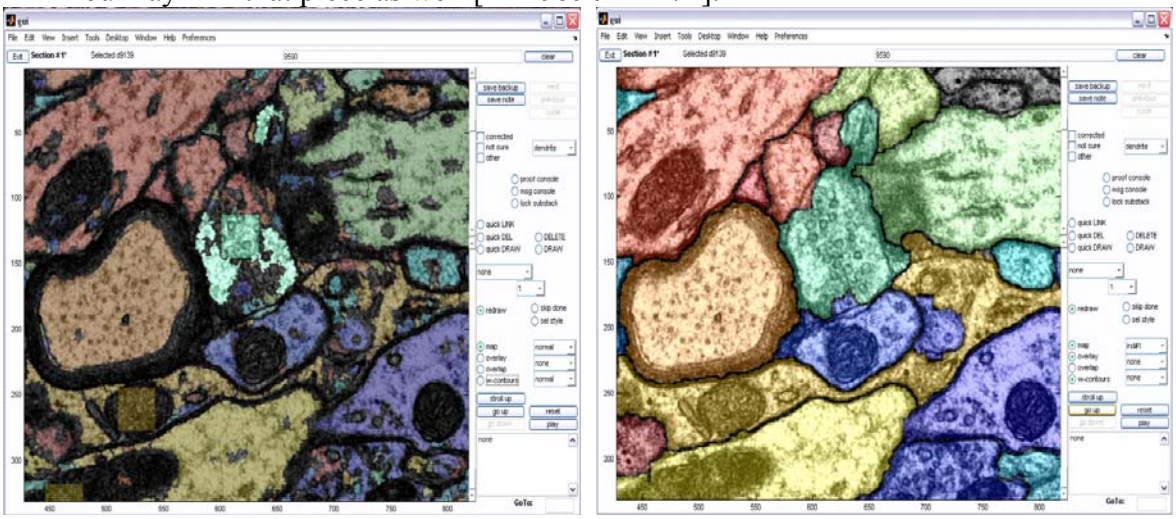


Figure 15 a9590 sufficiently well represented as is. Before watershed and after.

- #8216 is a glial process, for glia processes thin protrusions almost for sure are to be lost in post-processing. If this is of significance, either link enough pieces inside such thin protrusions to represent them, or draw size-1 squares along their centerlines. Once finished editing, press "3" to select type "glia", make sure "corrected" is checked and press "c" to cycle.

- #9172 is poorly represented axon, you may link one or two larger fragments, but most of the time you'll have to draw large squares over the "empty" space. Select "axon" as the type, make sure "corrected" is checked and press "c" to cycle, once done editing.
- #9845 is axon which is practically empty – while you may link some of its fragments, you will need to place approximately three size-8 squares approximately equidistantly over large mitochondrion in the center of this axonal bouton. See Fig. 15 for an example of how these draw markers should be placed. Browse through the entire stack and fix this blank area over the mitochondria similar to what shown in Fig. 15. Once done, select "axon", make sure "corrected" is checked and press "c".

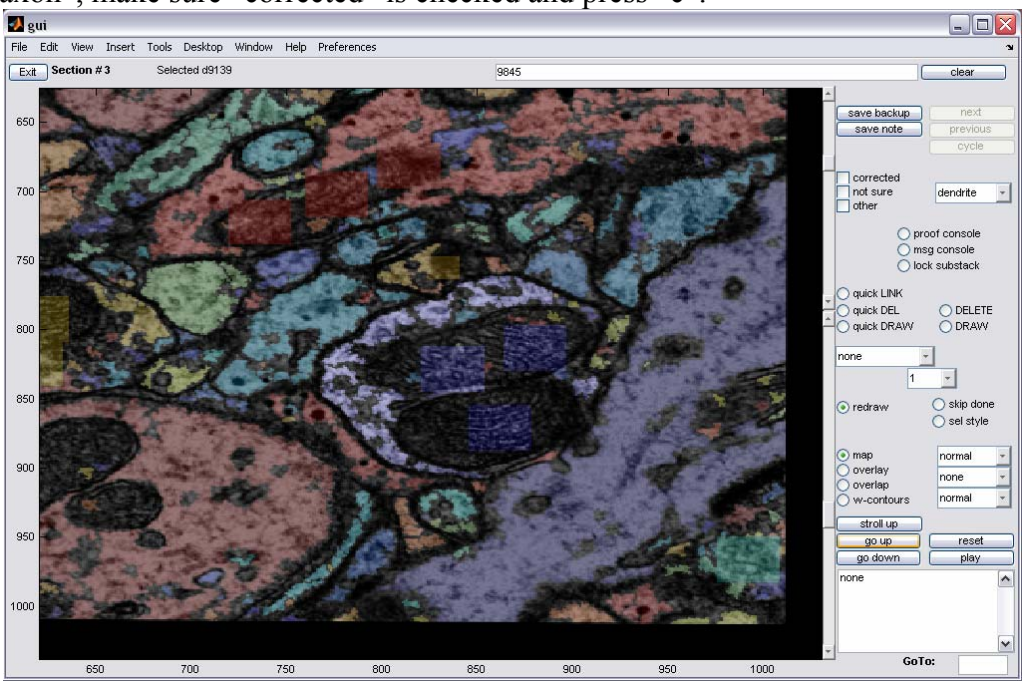


Figure 16 Place squares equidistantly accross large empty area in a9845.

- #1939 you will notice extends north, quick-link the fragments that belong to this axon. As you do so, you will notice that the axon will be outside of the field of view, use "move in" function on the right-click or "x" hotkey to re-adjust zoom when you link fragments so that you always have a view of the entire selected area. Alternate between original EM and the segmentation often using hotkey "q" to be able to get an unobstructed view at the situation. Once done editing, select "axon" for type, "corrected" and press "c".
- #7626 is large poorly segmented axonal bouton. First browse in one direction through the stack and quick LINK all "large" pieces, then browse in the reverse direction and place large square-markers over otherwise unmarked areas. In section 1 pay attention to bluish fragment to the west of the axonal bouton which extends inside a neighbor neuronal process. This is a situation where ASA merged parts of two objects in one. Because this error is not very extensive, quick LINK this object to #7626 and use DELETE operation with box of size 4 to erase part of the linked fragment that extends inside the neighbor cell. Then browse upward: you'll see that the link you created continued into the neighbor cell in that section. Select quick DELETE and delete wrong fragments. Browse upward and make sure that all errors propagated into the neighbor cell are fixed.
- For next item #3057, quick-link it to it a fragment in section 1 in the south of the axonal bouton. Once you browse one section up, you'll see, however, that this propagated into a large error with an axon passing on top getting connected to #3057. This is one scenario you want to rollback you editions. Use "ctrl-z" button for that. Once you cancelled this link, select quick DELETE and remove the problematic fragment in section 1. Other than that this axon needs no corrections, press "2" to identify it as axon, make sure "corrected" is checked and "cycle".

- #6295 is an oversegmented axonal bouton. Quick-link its peaces into the same object, in section 4 and 5 don't miss a larger segment extending south-west after a very dark mitochondrion. Use "flip" function ("q" hotkey) to get an unobstructed view at the EM. Also, place some draw markers in the centers of mitochondria and over severely oversegmented area, similar to some of the previous cases. Once you reached the uppermost section, browse downward to check you work. Pay attention to section 3 and 2, where you should notice a small offshoot at the south-west corner. If you've linked large pieces of that axonal bouton well enough, you'll be actually hinted at this by PRT Quick link and trace that piece downward. Once finished, press "2" to select "axon", make sure "corrected" is checked and press "c".
- #9393 is an axon which is fully located on the stack boundary. This object may be of no interest to us as it is both quite small and is only partially inside the volume. To have this object dropped from final proof, simply make sure that "corrected" checkbox is UNCHECKED and "cycle" on.
- #8223 is a glial process with many thin protrusions; again, it's up to you to decide to what precision you want to have it done. See Fig. 16 for the example of the pieces of this object. Once finished editing, press "3" to select "glia", make sure "corrected" is checked and "cycle" to the next item.

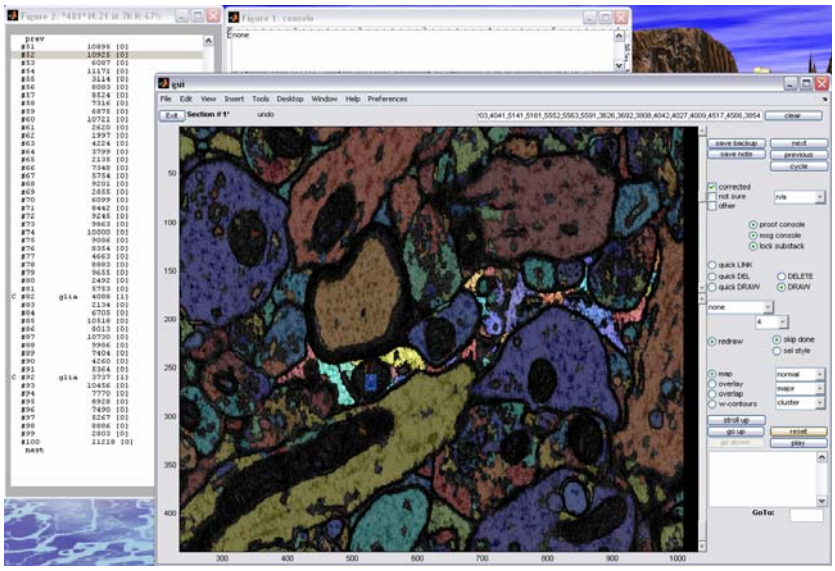


Figure 17 g8223 has many thin protrusions.

- #10469 has an offshoot to the east around sections 3-5, you'll be hinted at it by PRT GUI if you link its fragments sufficiently well; also, you may need to delete a small piece of this axon that penetrates inside the axon lying below it. Feel free to use "move in" or "x" hotkey function to re-adjust zoom to include fully a long axon you've just created.
- If you carefully trace #3114 south you'll note that it connects to a large piece of glial process g9384, that you should have attended to before. Once you link to it, a message in bold will be printed in notification area notifying you that you've linked to an object that had been before finalized. This means you don't need to go over just linked part once again, as you previously made sure it is correct. You will also see that the type of the whole collection of fragments should become "glia". Once you reached the point where you connected to already attended object and you are confident that the piece before that have been fully corrected, there is no more to do with this object – just click "c" to cycle to the next item.
- #8442 is a thin axon cut "in plane. It extends north-east and south-west often going "up" and "down" in a waving manner (see Fig. 17). Because this object is so thin and, thus, poorly imaged, it is important to "flip" often using "q" key to understand where it is going. It also pays off to first trace one of its ends (for example north-east direction), and

then the other (south-west direction). When you can't find where the axon went in the next section, try looking for it in the section in the opposite direction. Use general direction of motion to estimate position of the axon in the next section and look for it at that place specifically. If nothing works, look at the finalized example by selecting "w-contours" radiobutton. You can return back by deselecting "w-contours". Once finished, select "axon" for the type, make sure "corrected" is checked and "cycle".

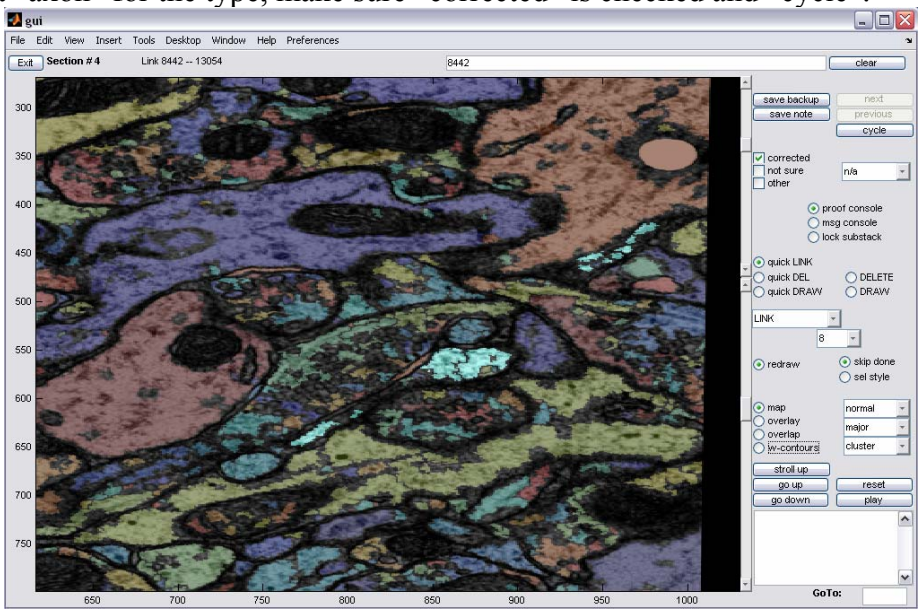


Figure 18 Pieces of axon 8442.

- #2173 when you place a square marker, not just this marker in the end is declared part of corresponding neuronal process, but also all segments that touch it. This provides a useful trick in case you have lots of smaller fragments which all together outline the shape of neuronal process well – just place square markers in a way that most of larger elementary blocks are touched. But also you need to make sure that the squares you place do not touched nearby objects.
- For next item #8886 pay attention that the axonal bouton to the east in sections 3-6 IS NOT its part.
- At this point the guided portion of this tutorial is over. We suggest that you continue with the rest of the proofing list continuing essentially the procedures above. You may refer for help for the finalized version of this stack at any time by selecting "w-contour" radiobutton.
- In the later part of the list you will run into large number of smaller pieces that belong to already finalized neuronal processes, or are partially covered by your marker squares, or are "could note be found" and, thus, are likely to be completely covered by your marker squares. As long as the backbone of the host neuronal process appears OK (and it should if you have verified this object earlier), you don't need to pay attention to such items. Just make sure that "corrected" checkbox is unchecked, so that they will be dropped from the final proof, and continue advancing through the list.

While you don't need to do this for the sample stack, whenever you will work on actual data you would need to create and maintain backup of your work. Backup is automatically saved for you every once in a while. Backup takes a snapshot of your workspace during proofreading session, so should you quit MATLAB and restart PRT, you should start from the point where the last backup had been saved. However, there are things automatic backup doesn't save, for example, final objects' contours. Full backup is saved using "save backup" button. Every time you leave your work place for an extended period of time we advise that you manually save backup by pressing "save backup" button.

If a backup had been saved for current dataset, when you start MATLAB next time, type "startgui" and select this dataset and click "start". A pop-up window will appear notifying you that a backup file had been found and asking whether it should be loaded. Click "Yes" to load backup and continue your work from the point where the backup had been saved. Be sure to select the same file you have been editing before – backups are data-file specific and, if you open a different file, PRT will not offer you to load the backup. Note that the state of the most proofreading controls will be reset – you will need to reopen proof console and set editing controls and proof-note conclusion boxes as necessary by hand.

The sample stack is a piece from actual Rat hippocampus data provided by Kristen Harris. It is about 10 cubic microns in diameter, it has about 150-200 final objects. It should take about 2-3 hours to complete in full.

Appendix B: PRT data files structure

At this time for the purpose of stand alone proofreading on 32 bit workstation ASA output for a stack of EMs is produced as a sequence of MATLAB mat files containing sequence of ***miniseries*** generally 10-20 sections long that represent together the full stack. Files are named following convention ***project_name.estackXXX-YYY.mat***. *Project_name* is a project-specific prefix; ***XXX*** and ***YYY*** refer to the absolute position (i.e. within the full stack) of the first and the last sections of the miniseries. Miniseries are generated to overlap over their two last/first sections, which is done to facilitate subsequent binding of miniseries in single continuous stack.

All files pertaining to single proofreading project should be kept in one directory. Proofreading process is organized in such a way that ASA data files will never be altered by PRT and all editions will be stored in separate files in the same directory.

PRT maintains two types of auxiliary data files: ***buff-files***, named as ***gui.buffXXX.mat***, and ***backup-files***, named as ***gui.dataXXX.mat***. GUI-loader ***startgui*** automatically filters out auxiliary files from the file list so that they generally will not be shown there.

gui.buffXXX.mat is automatically created to retain changes made in the miniseries starting at section XXX whenever operator browses out of this miniseries and loads an adjacent miniseries. This file is updated each time particular miniseries is unloaded. Likewise, this file is loaded and processed each time particular miniseries is loaded, so that, if operator navigates out of this miniseries and then returns, the miniseries will be reloaded with all editions that had been previously made there.

gui.dataXXX.mat file is the main backup container for corrections made by operator. It is created manually with “save backup” button and during every tenth automatic backup save. This file contains all information necessary to recover alterations introduced in the ASA segmentation by operator. This file is the main result of the proofreading work that will be fed into the subsequent stages of reconstruction pipeline. ***gui.data*** files are unique for each miniseries and are not overwritten when you advance proofreading process to the other miniseries.

Additional backup file, ***gui.backup.mat*** contains information about current GUI state and reference the current miniseries that is been edited. This file is used to locate your position in the proofreading list and otherwise change default GUI settings to facilitate smooth continuation of an interrupted proofreading session. This file is overwritten whenever you begin proofreading another miniseries; it is not miniseries-specific.

When PRT starts, PRT loads ***gui.backup.mat*** and checks if the miniseries specified there is the same miniseries that operator chose to start PRT with. If this is the case, a prompt will be shown offering operator to load the last backup file. If accepted, PRT will then load ***gui.dataXXX.mat*** and try to set GUI to the configuration specified in the ***gui.backup.mat***. The miniseries referred to in ***gui.backup.mat*** we call active miniseries. Only backup for active miniseries may be loaded at the time of PRT startup.

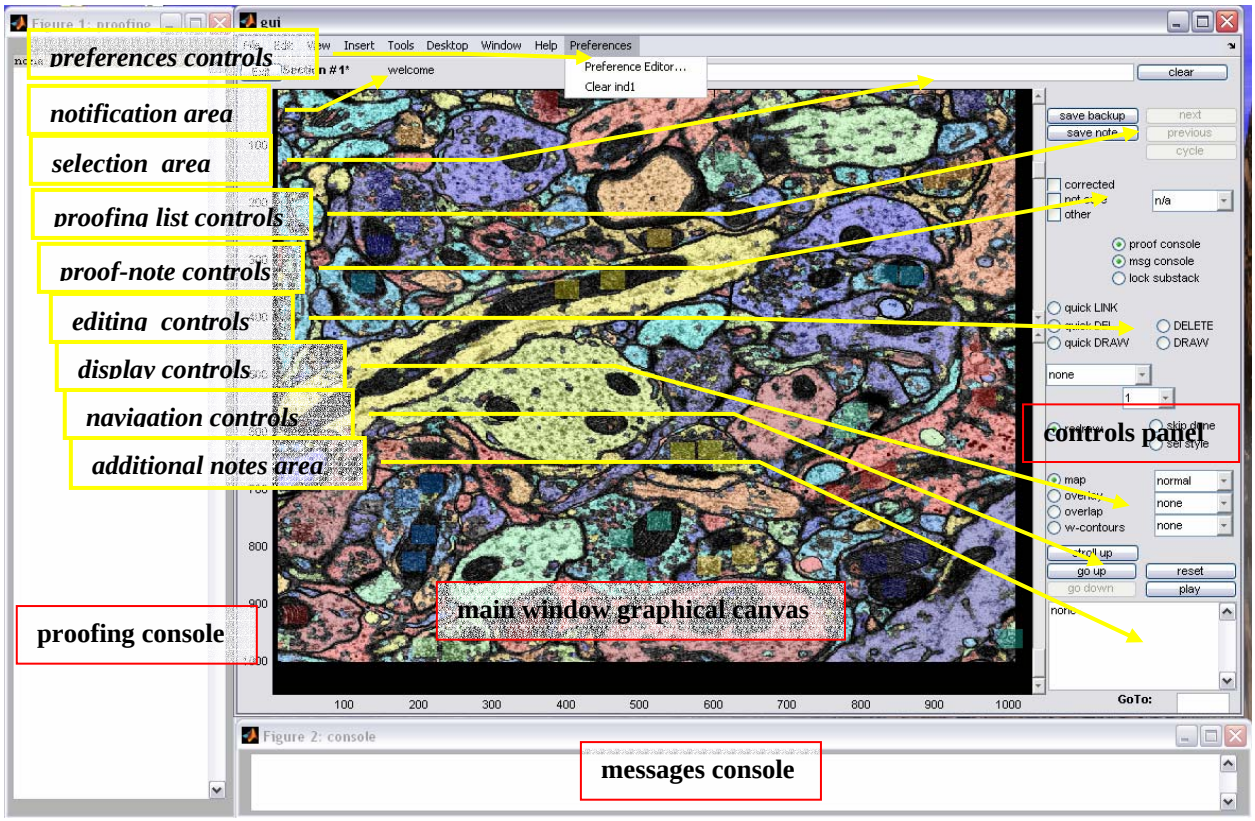
If miniseries identified in active backup does not coincide with the currently loaded miniseries, no prompt to load backup will be printed and backup file ***gui.dataXXX.mat***, even if present, will be ignored. To load backup file for miniseries which is not active, operator needs to use “load backup” command (see PRT GUI Controls section for more information) while within respective miniseries. Remember, however, that buff-file will still be loaded if found, thus, retaining all and any corrections made on the miniseries.

PRT GUI has option to either delete buff-files or retain them upon exit from GUI. If these files are retained, next time particular project’s data file is loaded editions recorded in the buff-files will be automatically applied regardless of whether the operator requested loading of the backup or not. If buff-files are removed, the data files will be loaded in their original form and operator would have to explicitly request loading of backup file to update the segmentation for each and any miniseries to include previous corrections. Any corrections not stored in respective ***gui.data*** files will be lost. We advise to always retain buff-files upon exit.

File ***pref.mat*** contains information about current state of preference options (see Preference Editor section for more information). This file is loaded and processed during PRT startup. By copying this file to a different project's folder, one may copy the preference settings configured earlier.

Appendix C: PRT GUI User Reference

Proofreading Tool GUI Controls



Graphical User Interface for Proofreading Tool can be essentially divided into following areas: ***proofing console, messages console, main window with graphical canvas***, which in turn subdivides into ***preferences editor, notification area, selection area, proofing list controls, proof-note controls, console controls, editing controls, display controls and navigation controls.***

Proofing console

Proofing console is used to display and interact with proofing list. It is activated with ***proof console*** radiobutton in the ***controls panel*** of ***main window***. When active, it lists items in the proofing list in the format **"C #40 glia 21929 [0] *comment*" :**

- "C" means that this object has a "corrected" proof-note and, thus, had been inspected at before, at some earlier time (not necessarily current proofreading session).
- "#40" stands for the item's number in the to-do list.
- "glia" (or dendrite, or axon, or extracell) is the object's type
- "21929" is numerical id within ASA segmentation.
- "[0]" is the flag which indicates whether the object had been inspected in *current* proofreading session (if "[1]"). As such, if you leave proofreading mode and re-enter it again, all "[1]" flags will be cleared, but all "C" flags will remain.
- "*comment*" indicates that the operator left a "not sure" proof-note for the object earlier.

The title bar of the proof console window shows proofreading stats in the following format: **"*3409* N:593 M:79K R:99%"**

- "*3409*" stands for the total number of items in the to-do list.
- "N:593" indicates how many distinct objects have been marked for the final proof.

- "M:79K" indicates largest size of the objects in the proofing list after the current one, in pixels.
- "R:33%" is an estimate of the volume fraction occupied by the objects that are currently marked for the final proof.

The menu bar of the proofing console allows for various Filtering and Sorting options to be performed on the proofing list. Filtering options include:

- "Major" restricts proofing list to the list of significant objects prepared by ASA (selected by default).
- "Checked" will restrict proofing list to those items that have "C" flag on them (and thus had been confirmed at any previous time).
- "Not Checked" will restrict proofing list to those items that do not have "C" flag on them (and thus had never been confirmed before).
- "Has Type" will restrict proofing list to those items that have a not "N/A" type assigned to them.
- "No Type" will restrict proofing list to those items that have only "N/A" type.
- "Not Sure" will restrict proofing list to those items that have a "not sure" proof-note associated with them.
- "Other" will restrict proofing list to those items that have a "other" proof-note associated with them.
- "By Slice" will restrict proofing list to items that are also referred to in particular slice. In case of "Not Sure" flag, e.g., this means items that have "not sure" proof-note saved in given slice. In case of "Lost Up", "Lost Down" flag, this restricts proofing list to itmes lost up/down in given slice. You may advance "current slice" set in the filter using "previous" and "next" buttons in the proofing list controls group.
- "By Type" will restrict proofing list to objects of particular types. Types selection is set in Preference Editor (see Preference Editor for details).
- "AUTO" will restrict proofing list to objects that have an "AUTO" proof-notes generated for them. This option is created for future use to identify warnings returned by automatic procedure binding proofread miniseries together.
- "Lost Up" will restrict proofing list to objects that had been lost in the subsequent slice somewhere in the stack.
- "Lost Down will restrict proofing list to objects that had been lost in the preceding slice somewhere in the stack.

Sorting options include:

- "Sort by Volume" will order items in descending order according to their volume.
- "Sort by Label" will order items in descending order according to their numeric segmentation label.

Other options include:

- "Skip Done" will force GUI to skip items with "[1]" flag (thus previously attended in current proofreading session) when using "cycle" function.
- "P-Map" will decrease proofing list by removing from it duplicates corresponding to same final label.

Proofing console allows you to browse, view and activate by selection the items from the proofing list. Note that whenever filtering or sorting options are changed, PRT will attempt to preserve selection to point to the same object. If currently selected object is not in the new filtered list, a closest position will be selected. If an item with "comment" is selected, all "not sure" proof-notes and their respective notes will be displayed in the ***messages console*** (below).

Messages console

Messages console is used to communicate to operator any additional messages. Currently, messages console is used to display content of comments associated with "not sure" proof-notes

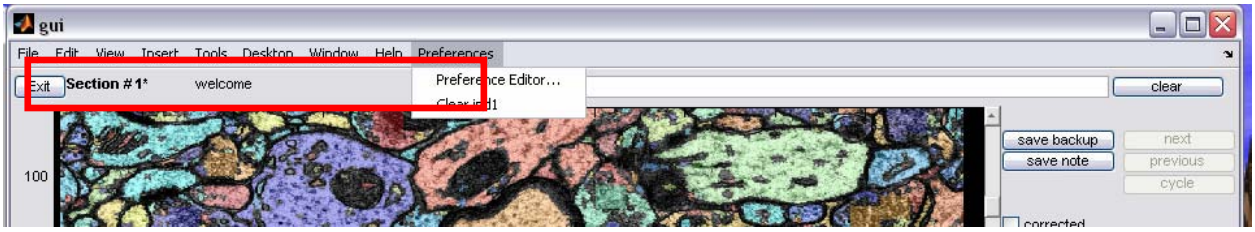
for an item selected in the proofing console, and to display messages about the progress of watershed post-processing procedure if operator requested “watershed” operation.

Main Window

Main window is primary GUI window, it contains ***graphical canvas*** on which PRT displays graphical information such as EM images or color-coded segmentation. Graphical canvas itself includes the canvas, two sliders to the right of the canvas and a dropdown menu.

- ***MATLAB main canvas*** is used to display EM images, color-coded segmentation and other graphical information and also to perform objects selection and order modifications of the segmentation. By default, clicks within the main canvas add objects to selection. During proofreading, clicks in the main canvas may also order modifications, as will be described in the section related to ***editing controls***.
- ***Mixer*** slider is the top slider to the right of the canvas and is used to control the mixing of EM images and color-coded segmentation, or any other two sets of images overlaid in the main canvas.
- ***Sections*** slider is the lower slider to the right of the canvas and is used to represent current section position in the miniseries and to navigate between sections.
- ***Dropdown menu*** can be called with left click of mouse in the main canvas and contains access to following functions:
    - ***move*** allows to re-center field of view around the point of the click;
    - ***zoom in*** allows to zoom in with a factor of 2 around the point of the click;
    - ***zoom out*** allows to zoom out with a factor of 2 around the point of the click;
    - ***move in*** is used to re-zoom around current selection.

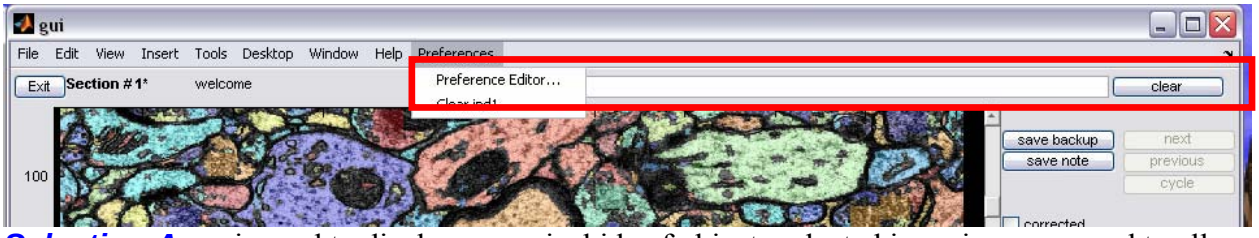
Notification Area



Notification Area is used by GUI to communicate short messages to the operator during PRT work. Notification area includes section counter and notification string.

- ***Section counter*** displays the number of currently displayed section. The number is the position within the entire stack from which miniseries had been pulled. Asterisk next to the number indicates that current section is either the topmost or the bottommost section of the miniseries.
- ***Notification string*** communicates short messages to the operator, among others, ***Selected ####***, ***Link #### - ####***, ***saving backup, undo*** etc. It will also show warnings such as ***Last edt at section ##***, ***type conflict***, etc. Important messages are displayed in bold.

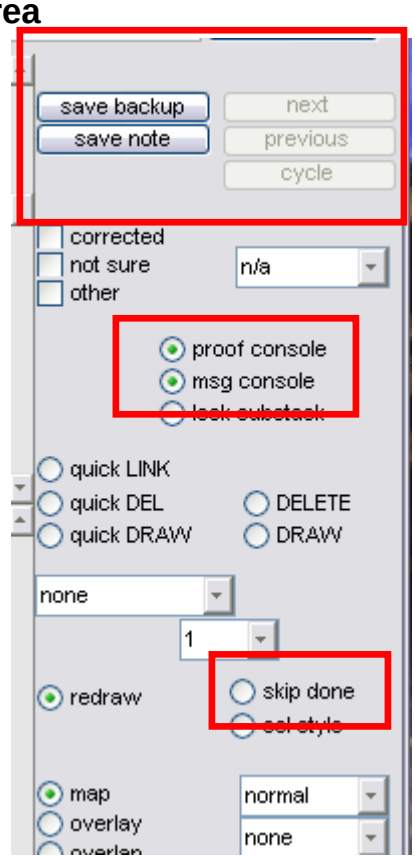
Selection Area



Selection Area is used to display numerical ids of objects selected in main canvas and to allow the operator to define selection manually via listing numerical ids of objects to be selected. List in the Selection area is comma separated. If “map” radiobutton is enabled in ***navigation controls***,

selection of objects is done according to their final label and final labels will be displayed in ***selection area***, if "map" radiobutton is disabled, forward labels are being selected and are displayed in ***selection area***. Selection area also includes ***clear*** button: this clears selection list as well as resets positions of other controls in ***controls panel***.

Proofing List Controls Area

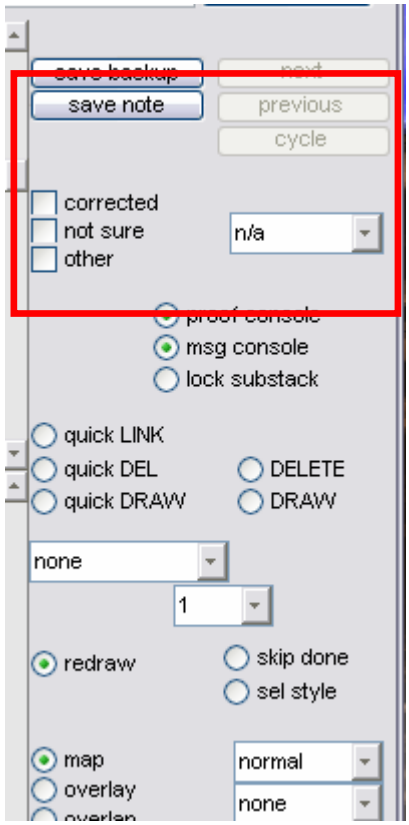


Proofing list controls includes ***next, previous, cycle*** buttons and ***save backup*** button and also ***skip done, proof console*** and ***msg console*** radiobuttons.

- ***Next*** button is used to shift "current slice" if "By Slice" filtering is selected for proofing list. This button is otherwise disabled.
- ***Previous*** button is used to shift "current slice" if "By Slice" filtering is selected for proofing list. This button is otherwise disabled.
- ***Cycle*** button is used to shift to next item in the proofing list, possibly, ignoring items related to previously inspected object, depending on the condition of "skip done" flag in proofing console, or "skip done" radiobutton if proofing console is not displayed. This is the recommended way of advancing through the proofing list.
- ***Save backup*** button is used to save backup file, including condition and current position in the proofing list. Backup file, when loaded, restore PRT GUI to the same condition where the backup file was saved except for the positions of editing controls.
- ***Skip done*** radiobutton, when enabled during proofreading mode, indicates that items related to previously inspected objects in the proofing list are to be skipped during "cycle". This is by default enabled when major proofreading mode is initiated and is disabled when major proofreading mode is disengaged. If proofing console is displayed, "Skip Done" flag from proofing console overrides "skip done" radiobutton.
- ***Proof console*** radiobutton is used to show/hide proofing console. Proofing console is used to display and interact with proofing list. See proofing console section above for more information.

- ***Msg console*** radiobutton is used to show/hide messages console. Messages console is used to display additional messages to the operator. See messages console section above for more information.

Proof-notes Controls Area

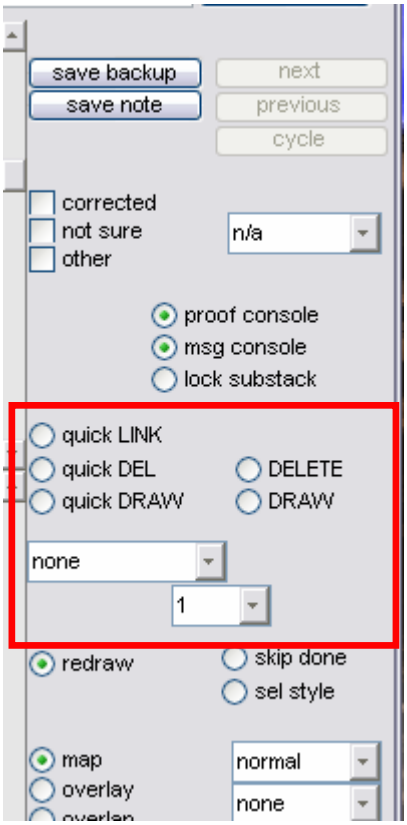


Proof-notes controls contain controls used state condition of inspected object in proofreading list, and create and save a proof-note. It contains ***save note*** button, ***corrected***, ***not sure*** and ***other*** checkboxes and ***types*** dropbox. This set of tools also contains ***additional notes*** area at the lower-right corner of GUI main window.

- ***Save note*** button is used to save created proof-notes. Proof-note is the means by which objects in segmentation are selected for the final proof. Final proof is composed of objects that run from the first section to the last uninterrupted and objects that have a "corrected" proof-note saved for them. All other objects, which do not have a "corrected" proof-note at all, or have a proof-note that has "not sure" flag and no "corrected" flag saved last, are dropped from the final proof. Final proof is used as the basis for watershed drawing of final contours.
    - For each process that appeared in proofing list and which should be kept in the final proof a "corrected" proof-note should be saved. Such note may be saved by checking ***corrected*** check-box and clicking ***save note*** button. Note that, when in proofreading mode ***corrected*** checkbox is automatically checked after the first correction is made by the operator. Also, initiated proof-note is saved automatically when operator shifts by any means to another item in the proofing list.
    - Operator may communicate with supervising person or leave notes to himself that can be accessed later by saving proof-note marked as "corrected" and "not sure" at the same time. Additional comments may be typed in the ***additional notes*** area: these comments are saved with the proof-notes and may be read at a latter time for "not sure" proof-notes. Additional notes are cleared each time operator shifts to new item in the proofing list.
    - Operator may instruct PRT to remove particular object from the final proof by saving only "not sure" proof-note for it. If both "corrected" and "not sure" proof-notes are

found for some object, the one saved the last will decide whether the object will be kept in the final proof.
    - Proof-notes, when saved, are associated with the last selected object and the current section, so that position of the event in question is referenced by these means. During proofreading, if default "corrected" state flag is set to "enabled" or "previous" in Preference Editor, proof-note is automatically initiated in the beginning of inspection of each new item from the proofing list with this item's id. When this happens and operator later checks also "not sure" flag, the section where "not sure" flag is written as the associated section for the proof-note. Otherwise, operator is responsible to make sure that appropriate object is selected and she is in the appropriate section for a "not sure" note. *We recommend, if operator decides to create a "not sure" proof-note long after she started editing an object, to uncheck all conclusion checkboxes and re-check them again in the section where confusing event occurred.*
    - When a proof-note is initiated, GUI displays an information message in notification area describing with which object's id and in which section this proof-note will be associated. This message in some cases may be promptly overwritten with following messages, e.g. in case of a "default" proof-note such message is always immediately overwritten by subsequent messages.
- Group of ***conclusion*** checkboxes, including ***corrected***, ***not sure*** and ***other*** checkboxes is used to state operator's conclusion about inspected object. "Corrected" conclusion means that the object had been examined and refers to a meaningful neuronal process in the final proof. "Not sure" conclusion means that operator has doubts about something related to this object in current section and wants this to be examined later. The content of operator's notes for "not sure" proof-notes may be viewed in "messages console" whenever an item with "not sure" proof-note is selected in the proofing list. Purely "not sure" proof-note explicitly indicates that the corresponded object should be excluded from the final proof. "Other" means that proof-note has also some other significance.
- ***Types*** dropbox is used to identify an object with either ***primary*** or ***secondary type***. ***Primary type*** refers to types such as ***dendrite, axon, glia*** or ***extracellular space*** – that identify objects that are independently present in the volume. ***Secondary type*** refers to types such as organelle, spine, protrusion, mitochondrion, etc. – objects that are usually part of larger object with certain primary type. A collection of fragments labeled with secondary type and linked to an object with some primary type is assumed to have that primary type.
    - Technically, type is associated with a forward label – this means that only objects carrying distinct forward labels may be identified with distinct primary or secondary types. When type is selected from ***types*** dropbox, selected type affects only currently selected *forward* label. Since primary type is defined by primary type of any one 3D piece in the collection representing particular neuronal process, one selection is sufficient to change the primary type of the entire collection. Similarly, when two objects are linked, primary type will propagate over the created link. If the two linked processes have different primary types, a warning message will be displayed to the operator in the notification area.
- ***Additional notes area*** textbox is used to save short messages with "not sure" proof-notes that can be viewed at a later time, or simply to make a note to oneself during proofreading. Each time proof-note is saved, content of this text-box is saved as well and the content of the text-box is cleared.

Editing Controls Area



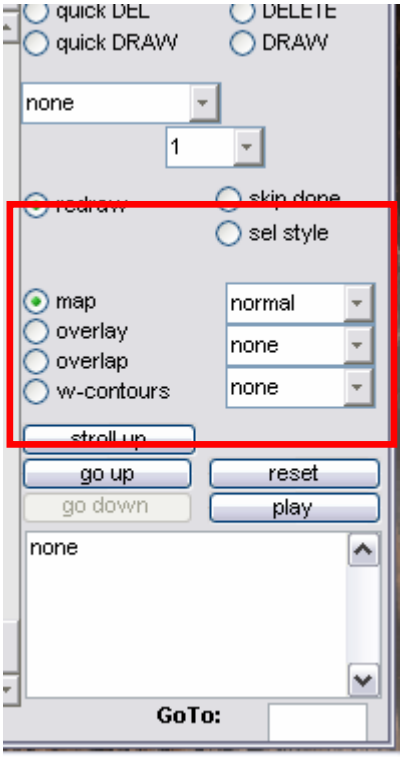
Editing controls contains tools that can be used to alter segmentation produced by ASA and are used during proofreading to fix errors made by ASA. This contains quick EDIT radiobuttons, operations dropbox and pen-size dropbox.

- ***operations*** dropbox contains list of all editing operations that GUI may perform on ASA segmentation output.
  - ***LINK*** operation is used to merge two separate objects in one. To perform link operation, select ***LINK*** and then click on the shape of the first object to be linked, and then click on the shape of the second object. Since link works on 3D objects, link operation will propagate automatically throughout all sections to segments connected to the linked objects. Link operation may be performed on two fragments in the same section as well as in different sections.
  - ***group*** operation is used to merge all objects currently selected in ***selection area*** into one object.
  - ***undo*** operation is used to undo last operation: simply select ***undo*** for that. Corrections made by operator are stored in ***undo*** buffer, as many as had been saved in the buffer may be undone. All operations listed in this section may be undone. Undo buffer is generally cleared when backup file is saved, but operator may adjust preferences so that the buffer is cleared each time she shifts to new item in the proofing list. See Preference Editor section for more information.
  - ***restore*** operation is used to restore object's original label assigned by ASA. This can be used to decouple a substantial group of fragments linked to another object by mistake and already finalized. The drawback of using ***restore*** is that all previous work and links that had been created for this object will be disrupted in an uncontrolled manner. Use of ***restore*** operation should be the last option only for substantially small objects, for which disruption introduced could be kept track of. Use of ***restore*** operation is subject to "know what you are doing" policy.
  - ***load backup*** operation is used to load last backup file. With this all changes made since last backup will be lost. This includes both delete/draw/recolor and link operations. See discussion of backup saves for more info.

- ***push id*** can be used to add an object with new id into the segmentation. This will change all internal arrays to accommodate a larger number of objects. Newly created id will be displayed in notification area so that operator can type it into selection area and draw with it, if necessary. Generally, this option is not used and if some object should be re-labeled with totally different ID, it is advised to take some of the many tiny disconnected objects nearby and use its id to perform any further editing. If needed, the original tiny object then can be deleted afterwards.
- ***watershed*** operation is used to draw watershed contours based on the final proof. Whether current section only or all sections in the miniseries will be affected is determined by appropriate setting in Preference Editor. See Preference Editor section for more information. This is a time-consuming and memory-consuming operation. You may use messages console to monitor ***watershed*** progress. After watershed finishes, it will set up GUI display to show watershed segmentation overlaid over original EM images. You may return to normal view by unchecking "w-contours" radiobutton in display controls, likewise, you may always display it again by checking "w-contours" radiobutton. Watershed segmentation is stored in ind#1 auxiliary internal variable and may be otherwise shown by setting display controls to display "ind#1" variable. See ***display controls*** section for more info. If OUT OF MEMORY errors appear, watershed contours may be purged using "clear ind1" command from ***preferences*** menu in the menu bar of the main window. Watershed segmentation is typically not stored during automated backup due to its large size. However, it is stored with ***save backup*** button.

- ***quick EDIT*** radiobuttons group is used to provide quick access to most frequently used editing operations. Only one ***quick EDIT*** mode can selected at a time, any other ***quick EDIT*** mode will be deselected upon click on any one of the ***quick EDIT*** radiobuttons.
  - ***quick LINK*** is used to speed up linking operation. In ***quick LINK*** mode only one click is required to establish association. If a proof-note had been initiated, ***quick LINK*** will be linking to the object identified in this proof-note as long as it is also part of the selection. If ***selection area*** contains only one object, ***quick LINK*** will be treated as linking to this object. If target of the quick linking is ambiguous, a message will be displayed in ***notification area*** and ***quick LINK*** will not be allowed to get enabled.
    Quick LINK is reset to a default state, identified in preferences, each time operator shifts to new item in the proofing list. It is recommended to have default "quick LINK" mode as "enabled".
  - ***quick DEL*** function is used to quickly ***DELETE*** entire elementary block from segmentation. ***Quick DEL*** uses either 4- or 8-connectivity to determine single-connected shape containing point of the click which will be deleted. 4- of 8-connectivity to use is set in the Preferences Editor.
  - ***quick DRAW*** function is used to quickly re-color entire elementary block to a new numerical label. ***quick DRAW*** is similar to ***quick DEL***. Drawing uses first numerical id listed in the ***selection area*** as the "color" of the pen.
  - ***DELETE*** operation is used to manually erase pixels colored with given label; removal of pixels is performed with pen of square shape of the size selected in the ***pen size*** dropbox.
  - ***DRAW*** operation is used to manually assign label to pixels; drawing is performed with pen of square shape of the size selected in the ***pen size*** dropbox. Drawing uses first numerical id listed in the ***selection area*** as the "color" of the pen.

Display Controls Area



Display Controls Area contains controls that allow the operator to manipulate graphical output of GUI as well as highlight selection. It contains set of radiobuttons, display modes drop-box, proofreading modes drop-box and selection modes drop-box.

- ***redraw*** radiobutton is used in conditions when graphical window update time is substantial [e.g. when editing large datasets or when working with large datasets on a server]. When ***redraw*** is deselected, GUI will not update screen each time the operator makes a correction. Only when ***redraw*** is again selected, the screen will be updated.
- ***map*** will force GUI to apply ***pmap*** either supplied by ASA or modified by the operator during proofreading. When ***map*** radiobutton is selected, all objects are colored with their final labels and each selection click will make selection of a final label. When ***map*** radiobutton is deselected, objects will be colored according to their forward labels and each selection click is selection of a forward label. This radiobutton is enabled by default at the time of the startup.
- ***overlay*** is used to display an overlay of two images, selected according to ***display modes*** dropbox. To display an overlay of two images, select first member of the overlay from ***display modes*** dropbox, then select ***overlay*** radiobutton, then select second member of the overlay from ***display modes*** dropbox. ***Mixer*** slider may be used to control mixing degree of the two layers. If ***display modes*** is in either “normal”, “forward”, “anchors” or “membranes” positions, choosing ***overlay*** will enable “special overlay” mode in which three EM sections (current, upper and lower) are overlaid in a color image with the lower section in RED, central section in GREEN and the upper section in BLUE channels. This special overlay mode is useful when trying to understand membrane motion.
- ***overlap*** mode, when used with most of display modes, shows overlap of objects according to display mode in current and the lower sections. When used with “slices” display mode, it will show min-overlap of grayscale intensities of two sections (i.e. pixel-by-pixel min of previous section and current section).
- ***display modes*** dropbox is used to select type of graphical information directed to main canvas. In ***compact mode***, set in preference editor, only part of this list is shown, which is relevant to all-major proofreading mode.

- in ***normal*** mode GUI displays overlay of EM images and color-coded segmentation.
- in ***slices*** mode GUI displays EM sections only.
- in ***labels*** mode GUI displays color-coded segmentation.
- in ***add#1*** modes GUI displays arbitrary grayscale stack stored in internal variable ***add1***. This variable may be accessed from MATLAB as ***global debug; debug.add1***.
- in ***ind#1*** mode GUI displays arbitrary color-coded stack stored in internal variable ***ind1***. This is a number-coded segmentation such as one produced by ASA. This variable may be accessed from MATLAB as ***global debug; debug.ind1***. ***watershed*** contour are stored in this variable after been drawn.

Following modes are not shown in ***compact mode***, additional data, typically not supplied with reduced 32 bit datasets, are necessary to display these:

- in ***forward*** mode GUI displays overlay of EM images and color-coded forward labels. Segmentations in "normal" and "forward" mode may differ both in color labels and segments outline due to specific of.
- in ***anchors*** mode GUI displays pixels recognized as "cytoplasm" inside the cells.
- in ***membrs*** mode GUI displays pixels recognized as membranes.
- in ***geom*** mode GUI displays profiles of all fragments – "elementary blocks".
- in ***forward*** mode GUI displays segmentation of forward labels only.
- in ***anchrs*** mode GUI displays pixels recognized as cell interiors only.
- in ***mmbrs*** mode GUI displays pixels recognized as membranes olny.
- in ***add#1-#3*** modes GUI displays arbitrary grayscale information stored as MATLAB 3D uint8 array in variable ***debug.add1-debug.add3***.
- in ***ind#1-#2*** mode GUI displays arbitrary color-coded information stored as number-coded segmentation MATLAB 3D array in variable ***debug.ind1-debug.ind2***.

- ***proofread modes*** dropbox is used to select type of proofreading pass to be performed by the operator.
  - ***none*** specifies no proofreading.
  - in ***major*** mode GUI will allow the operator to perform a pass following all-major strategy, in which the operator may systematically browse through all fragments in the stack considered as significant and then reconstruct neuronal processes containing such fragments in the entire volume of the miniseries at once.
  - in ***proof*** mode GUI will highlight all objects in the segmentation contained in the final proof version based on ASA output *and* proof-notes. Corrections may be made in ***proof*** mode to add or remove objects to the final proof or to alter objects, if they are found to be not properly represented.

  These options are not available in ***compact mode***:
  - in ***splits*** mode GUI will show the operator all split events.
  - in ***mergers*** mode GUI will show the operator all merger events
  - in ***losts*** mode GUI will show the operator all lost events.
  - in ***founds*** mode GUI will show the operator all found events.
  - in ***gather*** mode GUI allows operator to enter a series of points from the clicks performed in the main canvas. Points selected are displayed in console as **select (x,y,z) fragment id, click #** and are also stored in variable ***proof.pps*** as array ***(x,y,z,fragment id)***. ***Gather*** mode is complementary and may be used to enter a selection of points. Selection points are also accumulated in internal variable ***add1***. In current version of GUI this is disabled by default. To enable ***gather*** mode, set in ***data.allowgather*** variable to 1 when ***data*** is global.
- ***selection modes*** dropbox is used to set selection type used by GUI to highlight objects.
  - ***none*** mode means no-highlight is applied.

- ***normal*** mode is default context-dependent selection mode used with most display modes. In "normal" mode all selected objects are highlighted; in any proofread modes, all objects corresponding to events of selected type in current section are highlighted; in "proof" proofreading mode all major objects in the final proof are highlighted.
- ***cluster*** mode will highlight objects identified in the ***selection area*** regardless of the positions of ***display mode*** dropbox and ***proofread mode*** dropbox.
- ***filter*** mode is used with proofreading modes and shows all objects in the proofing console with current filter. This mode is not generally used with "all-major" proofreading mode.

These options are not available in ***compact mode:***

- ***event*** mode is context-dependent selection mode which highlights only fragments immediately participating in a selected ***ASA event***. This selection mode is used in some of proofreading modes when certain ***ASA log record*** is selected.
- ***same*** mode will keep current selection unchanged as operator browses up and down through the stack.

Navigation Controls Area

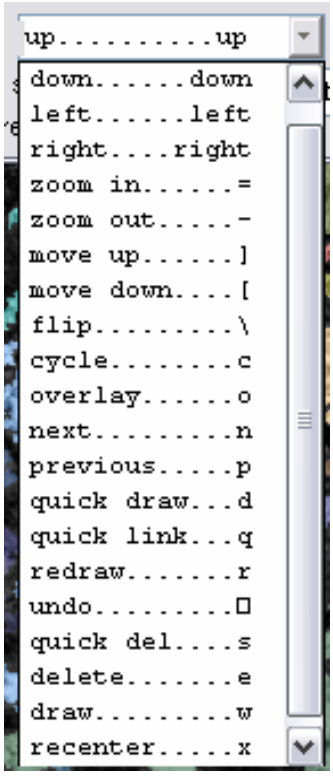
Navigation Controls Area includes navigation controls at the lower right of GUI as well as ***exit*** buttons, ***GoTo*** text-box and ***sel style*** and ***lock substack*** radiobutton.

- ***go down***/***go up*** buttons can be used to navigate through the stack in up/down direction.
- ***stroll up*** button can be used to "stroll" up to the last section where currently selected process is still found.
- ***reset*** button resets field of view to a full zoom.

- ***play*** button automatically browses through the stack in upward direction with current display and selection modes at rate of approximately three frames per second.
- ***GoTo*** text-box can be used to quickly go to specific section in the stack by typing in there section's number and pressing ***<enter>***.
- ***Exit*** button should be used to properly leave PRT GUI.
- ***Sel style*** radiobutton is used to alternate between two click-selection styles. By default, click-selection style is as this: left - selects, right - pop up menu, shift-left – deselect. Alternative is: left – selects, right – select box [click upper-left corner first and lower-right corner second], shift-left – deselect. Default selection style is set in the Preference Editor.
- ***Lock substack*** radiobutton prevents GUI from requesting adjacent miniseries and moving out of current miniseries when you accidentally venture out of the current miniseries during proofreading. GUI allows navigation out of the current miniseries into adjacent miniseries, particularly for the purpose of resolving ambiguous events near the bottommost or topmost sections of the miniseries. However, to prevent time-consuming accidental reloading of miniseries during regular proofreading, we advise to have ***lock substack*** radiobutton enabled at all times unless you need to leave current miniseries to resolve an ambiguity in upper or lower sections.

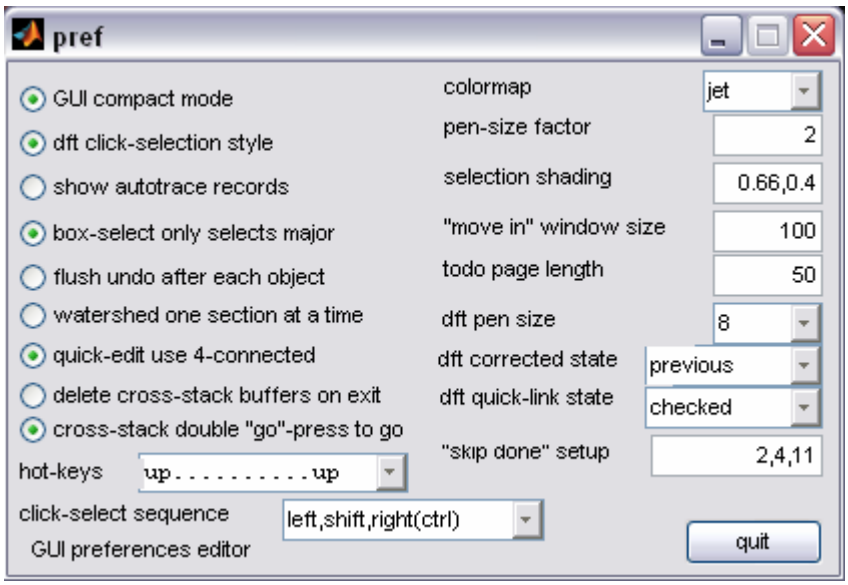
GUI Hotkeys Functions and Preference Editor

GUI supports calls to following functions via hotkeys. Listed below are default settings:



- ***up/down/right/left*** cursor keys can be used to move field of view in these directions.
- **=/-** can be used to zoom in/zoom out around current center of the field of view.
- **[/]** can be used to browse down/up through the stack.
- **** key can be used to quickly switch between current display/selection mode and EM images [***flip*** mode].
- **c** is hotkey for ***cycle*** button.
- **o** is hotkey for overlay radiobutton.
- **n** is hotkey for ***next*** button.
- **p** is hotkey for ***previous*** button.
- **d** is hotkey for ***quick DRAW***.
- **q** is hotkey for ***quick LINK***.
- **r** is hotkey for ***redraw***.
- **ctrl-z** is hotkey for ***undo***.
- **s** is hotkey for ***quick DEL***.
- **e** is hotkey for ***DELETE***.
- **w** is hotkey for ***DRAW***.
- **x** is hotkey for ***re-center & re-zoom***.
- **1,2,3** are hotkey for ***type selection*** for dendrite, axon and glia, respectively.

Hotkeys assignments may be changed via ***Preference Editor*** accessible through ***Preferences*** submenu of the GUI main menu.



Other preferences adjustable via ***Preference Editor*** are:

- ***GUI compact mode*** tells GUI to start in compact mode. This takes effect during next GUI startup. ***Compact mode*** hides some of the options from ***display mode*** dropbox and ***proofreading mode*** dropbox that are not used with all-major strategy.
- ***dft click-selection style*** whether default selection style is “select/deselect/pop up menu” or “select/deselect/box select”. Such implementation is due to mouse-events-recognition limitation of MATLAB which only distinguish three types of mouse click events.
- ***show autotrace records*** tells GUI whether it should show or hide auxiliary ASA debugging records. These are generally not supplied with 32 bit miniseries and enabling this radiobutton should have no effect.
- ***box-select only selects major*** whether during “box-select” only those objects that are also present in the list of significant object prepared by ASA should be selected, or any objects in ASA segmentation.
- ***flush undo after each object*** whether undo buffer should be cleared each time operator begins editing new item in the proofing list. Otherwise undo buffer is only cleared when backup is saved.
- ***watershed one section at a time*** whether PRT should apply watershed post-processing only to current section or to entire miniseries when “watershed” operation is requested.
- ***quck-edit use 4-connected*** whether quick DRAW and quick DEL should use 4- or 8-connected definition of connectedness.
- ***delete cross-stack buffers on exit*** whether PRT should delete miniseries buffer files, used to keep track of operators modifications when operator browses across miniseries boundaries. Leaving buffer files will cause PRT to load miniseries at startup with all and any modifications previously made, *even if load backup request was denied*. Removing buffer files will cause PRT load miniseries in their original form, except perhaps for the backup file saved for current section, if permitted. However, PRT will never remember correction made otherwise during previous MATLAB sessions. We advise to have this option ***enabled*** unless you are certain you know what you are doing.
- ***cross-stack triple “go”-press to go*** whether GUI should wait for user to press ***up*** or ***down*** three times before requesting adjacent miniseries – enabled to reduce chance to accidentally venture out of the current miniseries. Also, to prevent time-consuming reloading of miniseries during proofreading, have ***lock substack*** radiobutton enabled at all times unless you need to leave current miniseries to resolve an ambiguity in upper or lower sections.
- ***colormap*** defiles the colormap used in graphical canvas to translate numerical segmentation labels into color image.

- ***pen-size factor*** is the size of size-1 square pen, used for drawing, in pixels.
- ***selection shading*** defines by how much "unselected" part of image will be shaded. This is a two-number field, first number applies generally while second number is used in "proof" proofreading mode.
- ***"move in" window size*** defines size of margins on the side of currently selected object that will be added to the zoom when "move in" operation is requested either via pop-up menu or keyboard.
- ***todo page length*** defines how many items are shown in one page of proofing list in ***proofing console***.
- ***dft pen size*** defines the state of the pen-size dropbox set by default when operator begins editing new item from the proofing list.
- ***dft corrected state*** defines the state of the ***corrected*** checkbox set by default when operator begins new item from the proofing list.
- ***dft quick-link state*** defines the state of the ***quick LINK*** radiobutton set by default when operator begins new item from the proofing list.
- ***"skip done" setup*** defines types of the objects selected when "By Type" filter is applied to proofing list. Numerical references are according to positions in ***types*** dropbox. By default, ***"2,4,11"*** means dendrites, glia and other. Generally, these are objects that are not warranted to be correct (due to spines, glial protrusions etc.) even if they have a "confirmed" flag on them.
- ***hot-keys*** defines assignment of hotkeys for commonly used functions. To assign new hot-key for a function, select this function from the dropbox (a message in bold "press key associated with this function" should appear at the bottom of the Preference Editor) and then press desired keyboard key. Hotkeys are prioritized as they are listed in the dropbox, thus if two functions are assigned to the same key, the one higher in the list will be actually executed.
- ***click-select sequence*** defines assignment of clicks to "select/deselect/{pop-up menu or box-select}"

Preferences are saved upon pressing "quit" button. Preferences are saved into ***pref.mat*** file; this file is loaded and applied automatically during PRT startup. To carry over your preferences from one project to the other, simply copy this file ***pref.mat*** into the folder containing ASA data files for this project.