

Larva Behavior Labeling Tool

Steve Branson
sbranson@cs.ucsd.edu

Kristin Branson
bransonk@janelia.hhmi.org

September 14, 2010

1 Introduction

The larva behavior labeling tool is a tool for annotating, training, and detecting larva behaviors and annotating and fitting larva skeletons. The basic tool is depicted in Figure 1.

2 Installation and Compilation

The tool was developed and tested mostly on Ubuntu 9.10 64-bit Linux, and had been compiled and run (but not thoroughly tested) on Windows 7 and Mac OS X Snow Leopard.

2.1 Running the Tool

Obtain the package by emailing Steve (sbranson@cs.ucsd.edu), or by downloading using SVN (see 2.2.1). The package should contain a folder called *larva* with a bunch of subfolders (like *code*, *doc*, *data*, *windows*, *mac*).

- **Windows:** Double click the file *gui.exe* in the folder *larva/windows*
- **Mac:** Double click the file *run* in the folder *larva/mac*
- **Linux:** From a terminal, type *./gui* from the folder *larva/code*

If this doesn't work, please email Steve (sbranson@cs.ucsd.edu) with a description of what happened and any error messages that appear. It is highly possible that I forgot to include some necessary libraries to run on some versions of for Mac or Windows.

2.2 Running and Compiling From the Source Code

You can also obtain the tool's source code and use that to compile an executable and run the tool. This is currently the preferred method of running the tool, because I am still making changes to it and you will be able to run the latest version of the tool.

2.2.1 Downloading the Source Code Using SVN

SVN is a revision control system that makes it easier for multiple developers work on the same code, by tracking and maintaining different versions and changes to the code.

- **Step 1:** Check out the SVN repository. **This is a one time thing**, and is basically how you download all the larva code and data initially. If you have done this already, proceed to Step 2.
 - **Mac:**
 - * Open a terminal (located in /Application/Utilities).

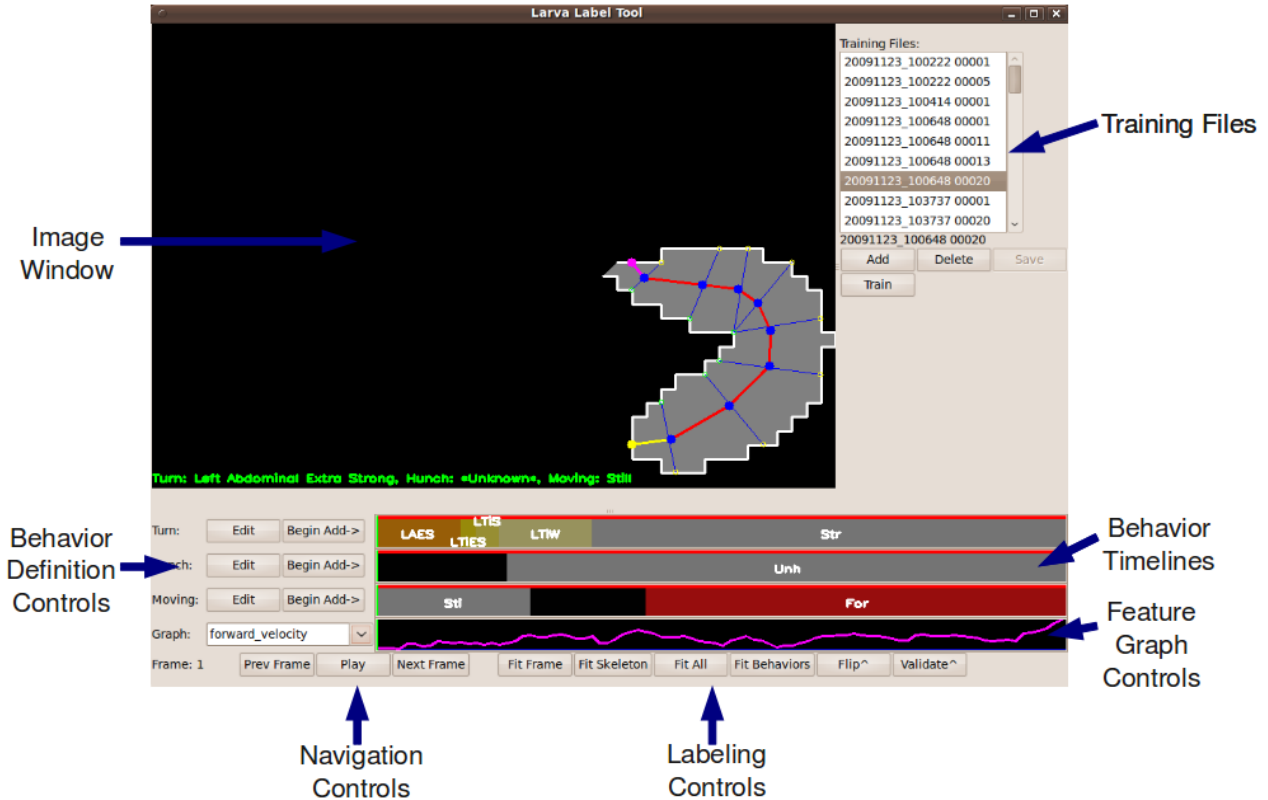


Figure 1: Screenshot of the Larva Behavior Labeling Tool

- * Navigate to the folder where you want to put the code (you can type the command `cd folder_name` to enter the directory `folder_name`).
- * Check out the repository by typing
`svn checkout https://subversion2.int.janelia.org/branson/projects/larva` (you must be connected to the Janelia network). If it succeeds, it should create a directory called `larva`, and you should see it print out a list of files as they are downloaded.
- * If you get an error to the effect of “svn doesn’t exist”, you can install svn by following Steps 1-3 at <http://www.wikihow.com/Install-Subversion-on-Mac-OS-X>
- **Windows:**
 - * If you don’t have an SVN client, you can install Tortoise SVN: <http://tortoisesvn.net/downloads>.
 - * Use a file browser to go to the directory where you want to put the code.
 - * Right-click, you should see a Windows menu popup. Select the menu option *SVN Checkout*.
 - * In the field *URL of repository* enter
`https://subversion2.int.janelia.org/branson/projects/larva` (you must be connected to the Janelia network), then press *Ok*. If it succeeds, it should create a directory called `larva`, and you should see it print out a list of files as they are downloaded.
- **Linux:** Same as Mac, except that if you don’t have SVN, install the package `subversion` (e.g., `sudo apt-get install subversion`)
- **Step 2:** Get the latest version of the code. *Note:* you must be connected to the Janelia network.
 - **Mac:** From a terminal (located in /Application/Utilities), navigate to the folder ‘larva’ (e.g., `cd folder_name`), which was created using the previous step. Type `svn update`

- **Windows:** From a file browser, right-click on the folder *larva* (created using the previous step). A windows menu should popup, select *SVN Update*
- **Linux:** Same as Mac

2.2.2 Compiling and Running the Source Code

After following the steps in Section 2.2.1, you should have a folder names *larva* and can compile the code into an executable. If it works it's pretty easy. If you get errors, email Steve (sbranson@cs.ucsd.edu) with a description of what happened and any error messages that appear.

Note: The code has 2 main library dependencies: wxWidgets and OpenCV. For Mac OS X, I tried to package these libraries inside *larva/mac_local*, which worked on Wyatt's laptop; however I'm not 100% sure it will work on other computers.

- **Mac:**

- Open a terminal (located in /Application/Utilities).
- Navigate to the folder *larva/code* (e.g., type `cd larva/code`)
- Set the library path by typing `export DYLD_LIBRARY_PATH=./mac_local/lib`
- Type `make` to compile the code. If it succeeds, you should see a bunch of lines printed like `g++ -g -Wall...`, and it will generate an executable file *larva/code/gui*.
- Run the larva labeling tool by typing `./gui`

- **Windows:**

- Open the project file */larva/code/larva.sln* (Requires the Microsoft Visual C++ compiler)
- Make sure you have installed OpenCV <http://opencv.willowgarage.com/wiki/VisualC%2B%2B> and wxWidgets (http://wiki.wxwidgets.org/Microsoft_Visual_C%2B%2B_Guide)
- Select the menu option *Build* → *Build Solution* to compile the code. If it succeeds, it should print a line `===Build: 1 succeeded` in the output log, and it will generate an executable file *larva/windows/gui.exe*.
- Open the file *larva/windows/gui.exe* to run the labeling tool

- **Linux:** Same as Mac. Make sure you have wxWigets

(`sudo apt-get install build-essential libwxgtk2.8-dev libwxgtk2.8-dbg`) and OpenCV (<http://opencv.willowgarage.com/wiki/>)

3 User Guide

3.1 Opening/Importing/Removing Files

- **Opening a File:** Open a larva file for annotating by clicking an entry in the *Training Files* listbox.
- **Importing a File:** To add a file not yet in the *Training Files* listbox, click the *Add*, browse for a .outline larva contour file (as outputted by Rex's tracker), then click *Open*. The file should be added to the bottom of the *Training Files* listbox. Files inside this list will be used to train the Behavior Detector.
- **Removing a Training File:** To remove a file from this list, click the *Delete* button. This removes the file from the training file list, but does not physically delete the file.

3.2 Detecting Behaviors and Fitting Skeletons

Once you have opened a larva file (Section 3.1), you can try to automatically fit a skeleton to each larva contour in the tracked sequence and detect the corresponding sequence of larva behaviors. This is done using the Fit buttons in the *Labeling Controls* panel. The skeleton is a curve passing through the center of the larva from the tail to the head. It is depicted as a series of red line segments in the *Image Window*. There are 3 predefined groups of behaviors: *Turn*, *Hunch*, *Moving*. Each behavior group is displayed and labeled independently (there are 3 different timelines in the *Behavior Timelines*). Each behavior group has a collection of mutually exclusive behavior classes, e.g. *Left Thoracic Weak*, *Right Tip Strong*, *Straight*, etc. The behaviors are depicted in the *Behavior Timelines* as a sequence of colored bars (see Figure ??).

The *Fit All* button will detect both the skeleton and the behavior sequence, whereas *Fit Skeleton* and *Fit Behaviors* will detect only one or the other. You cannot detect behaviors unless a detected skeleton already exists (either using *Fit Skeleton*, *Fit All*, or stored in an existing annotation file if you have edited this larva file before). You should be able to see the results of a fit skeleton and/or behavior sequence in the *Image Window* and *Behavior Timelines*.

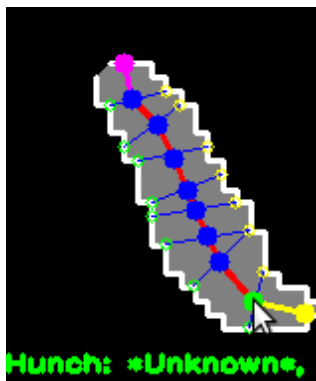
3.3 Viewing and Navigating Around Larva Blob Sequences

Once you have opened a larva file (Section 3.1), an image should appear in the *Image Window* which shows the larva blob for frame 1 of the tracked blob sequence. A tracked blob sequence corresponds to a particular larva animal over some period of time $t = 1...T$, as returned by Rex's tracker. The *Behavior Timeline* controls are used to represent this period of time, where the leftmost part of the timeline corresponds to the first frame $t = 1$ and the rightmost part corresponds to $t = T$. Hovering the mouse over a portion of the timeline will draw the larva blob of the corresponding time in the *Image Window*. Clicking a point in the timeline will set the current frame to the corresponding point, visualized using a vertical green bar (see Figure 2b). The buttons *Prev Frame* and *Next Frame* will change the current frame to one frame earlier and one frame later, respectively. Pressing the *Play* button will play the blob sequence in real time. You will need to press the *Play* button again to pause/stop the play animation.

3.4 Annotating Behaviors

To add a new behavior bout, follow the steps outlined in Figure 3.4. Press the *Save* button near the *Training Files* list to save changes to the currently opened larva blob sequence.

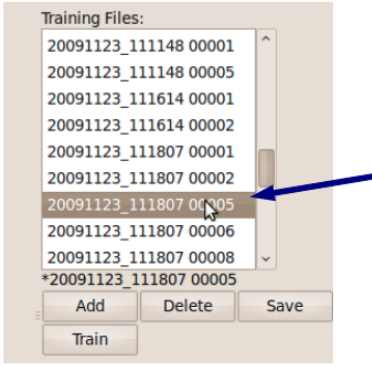
3.5 Annotating Skeletons



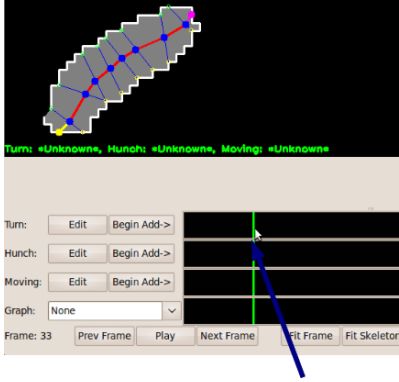
In general, you should just use the automatic skeleton fitting routines (Section 3.2); however, it is also possible to adjust the skeleton location manually. This can be done by clicking and dragging one of the blue skeleton points in the *Image Window*, which should appear in green when you place the mouse over it. Press the *Save* button near the *Training Files* list to save changes to the currently opened larva blob sequence.

3.6 Verifying Annotations

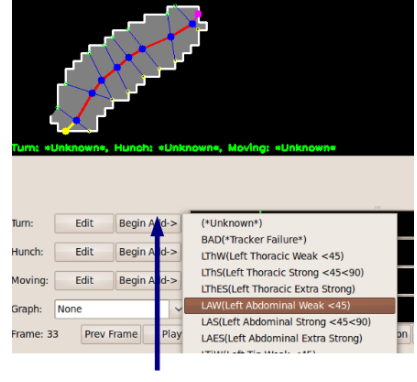
Since it is possible for the computer to automatically predict both the skeleton and the behavior, a human has to verify the correctness of the predicted skeleton/behavior before that data can be considered to be good training data (for training a better behavior detection algorithm). A frame is automatically marked as verified if it was labeled manually. The procedure for verifying a sequence of frames as correct is similar



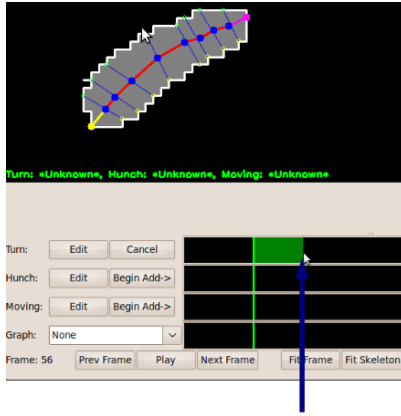
(a) Select a file to annotate by clicking an entry in the *Training Files* listbox.



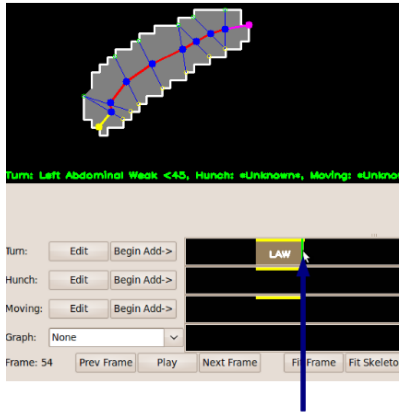
(b) Click on the *Behavior Timeline* at the time frame when your behavior starts.



(c) Click on the *Begin Add* button next to the behavior group you want to add a behavior for (in this case *Turn*).



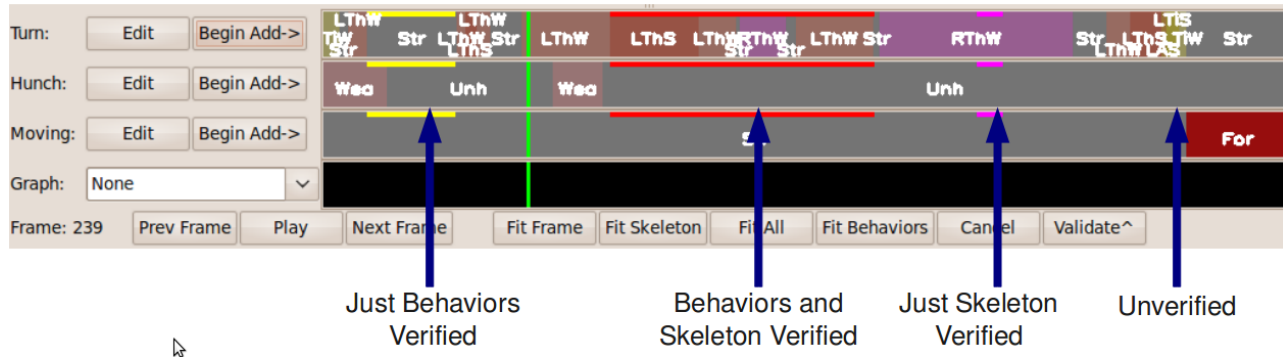
(d) Move your mouse over the *Behavior Timeline* to the frame when your behavior ends. A green highlighted region should appear.



(e) Click the mouse to finish adding your behavior. The new behavior should appear on the timeline.

Figure 2: **Adding a New Behavior Bout:** The example above depicts the sequence of steps used to add a new behavior bout for *Turn Left Abdominal Weak*.

to the procedure for labeling behaviors in Figure 3.4, except that the *Verify* button is used instead of the *Begin Add* button. In other words, click on the timeline the starting frame of the sequence you want to verify, then click the *Verify* button, then click on the timeline the ending frame of the sequence you want to verify. Verified frames will be displayed with a bright red strip on the top of the timeline. Frames where the skeleton has been verified but not the behavior are drawn in magenta, and frames where the behavior has been verified but not the skeleton are drawn in yellow. Only bright red frames (where both the skeleton and behavior have been verified) are used for training. Press the *Save* button near the *Training Files* list to save changes to the currently opened larva blob sequence.



3.7 Training the Behavior Detector

After you have annotated some larva blob files, you can train a behavior detector by pressing the *Train* button near the *Training File* listbox. The learning algorithm will learn from any frames in files in the training list that have been verified and highlighted in bright red (see 3.6). Training may take on the order of hours to complete. After training finishes, the *Fit Behavior* button can be used to apply the learned detector to a larva blob sequence. Some tips on constructing a good training set are:

- Try to label at least 10-20 larva files
- Make sure the larva files that you annotate are representative of the larva files that you want the system to work on. In other words, if the annotated larvae come from a different experiment where the larvae appear very different, the learned behavior detector won't behave very well.
- Make sure you get a lot of example sequences for every type of behavior. For example, only having one annotated bout of some rare behavior (e.g. *Sigmoidal left to right*) might cause that behavior detector to be very unreliable. It is best to have at least 10-20 labeled examples of each type of behavior.
- If possible, label or verify all frames in a given larva blob sequence (as opposed to verifying some frames in a particular file while leaving some other frames unverified)
- There are 2 special "behaviors" added by default *Unknown* and *Tracker Failure*. The label *Unknown* pretty much means "don't care" or "can't tell", e.g. it doesn't really matter what these frames are labeled as. The label *Tracker Failure* should be used when the larva blob contour was bad (e.g. the shape of the detected blob is very different from the shape of a real larva, due to a tracking problem), such that the skeleton fit was bad.
- If you train on a machine with multiple processors/cores, it will run faster

3.8 Changing the Behavior Definitions

You can alter the list of the types of behaviors, e.g. *Turn Left Tip Strong*, either adding new behaviors, deleting old ones, or renaming old ones. On the other hand, you should be careful about doing this because behavior annotation files depend on the defined list of behaviors such that changing the list of behaviors may invalidate them. Any files relating to learned behavior detectors will almost certainly be invalidated (you may need to manually remove files in *larva/data/classifier* to avoid errors). As a result, it's a good

idea to just edit behaviors once, before you start annotating files, and then not change them again. If you still want to edit the behavior definitions, it's a good idea to backup your files first (see Section 3.9).

You can edit behaviors by clicking on the *Edit* button in the *Behavior Definition Controls*, which should open up a window with the list of available behaviors. Use the *Add*, *Edit*, *Delete* buttons to alter the list of behaviors as desired. It's pretty straightforward what these controls do. The one non-obvious option is the *Color* field when editing a particular behavior. This is the color that is used to draw the behavior in the timeline controls, defined in hexadecimal 0xRRGGBB, where RR is the red component, GG is the green component, and BB is the blue component (see <http://www.computerhope.com/htmcolor.htm>). Any changes you make will not take effect until you select the *Save* button.

There is currently no mechanism in the GUI for defining new Behavior groups (e.g. Turn, Hunch, Moving); however, you can probably do this by editing the file *larva/data/behaviors.txt*. Changes made to this file should show up in the GUI.

3.9 Backing Up Files

It might be a good idea to backup files before you make any changes to the behavior definitions or training file list, since changing the behavior definitions can invalidate existing annotations. You can backup all configuration and annotation data files (all files described in Section 4) by clicking the *Backup* button under the *Training Files* listbox. This will create a directory *larva/backup/backup_timestamp*, where *backup_timestamp* is the current time/date. You can run the label tool GUI using the backed up files (this will both read in the backed up files and save new annotation changes to the backed up files) by running starting the label tool with the command line arguments: `./gui -d ../backup/backup_timestamp`.

4 Data Files

Data files are stored in the folder *larva/data*. The most important configuration related data files (you should consider backing these up before you make any major changes) are:

- *larva/data/train.txt*: contains the list of larva blob data files that are used for training the behavior detector and are selectable *Training Files* listbox in the GUI. This file gets edited when you Add/Delete files in the *Training Files* listbox.
- *larva/data/behaviors.txt*: contains the definition of behaviors that are detectable and annotatable. These are editable in the GUI in the *Behavior Definition Controls* and annotatable in the *Behavior Timelines*. This file gets edited when you Add/Edit/Delete behaviors in the *Behavior Definition Controls*.
- *larva/data/classifier/All.svmbehavior*: contains the data files for the learned behavior detector. This is generated when the user trains the behavior detector via the *Train* button.

The data files containing annotations for individual animals are stored in the folders like *larva/data/20091123_100222*. These are direct copies of data folders that were generated from Rex's Multi Worm Tracker. The Larva Behavior Tool imports files like *larva/data/20091123_100222/20091123_100222@w1118@n@t1@p_strongside_15s1x30s0s#n#n#n@1@.00001.outline*, which contain the contours of tracked larva blobs and are generated by Rex's tracker. It outputs files like *larva/data/20091123_100222/20091123_100222@w1118@n@t1@p_strongside_15s1x30s0s#n#n#n@1@.00001.anno*. The *.anno* files contain all information stored in the *.outline* files, as well as information relating to the detected or annotated larva skeleton, and detected or annotated larva behaviors.