This package of Matlab and FIJI scripts is provided as the source code used to process data presented in the eLIFE article titled “Axon guidance receptor Frazzled promotes growth cone attachment at the source of a Netrin gradient in the *Drosophila* visual system.” by Orkun Akin and Larry Zipursky. In addition to custom scripts written by the first author, the package includes software contributed to the Mathworks File Exchange repository under the BSD license by the following individuals: Christophe Bernard, Tolga Birdal, Dan Couture, John D'Errico, Matt Fetterman, Damien Garcia, Levente Hunyadi, Dirk-Jan Kroon, Jose Vicente Manjon-Herrera, Nima Moshtagh, Dylan Muir, Francois Nedelec, Dirk Padfield, Yury Petrov, Roland Pfister, Val Schmidt. We thank them for sharing their work.

The suite of scripts in the ‘MainScripts’ folder are designed to take a raw 4D data series of R8 development and produce as the final output the individually registered growth cone panels shown in Fig.1e and other figures of the article cited above. These scripts are written for an expert user by a non-expert programmer; they are lightly commented, rigid in their expectation of input arguments, and they contain commented-out ‘relic’ lines of code. While this text will be providing instructions on how to move the data through this suite, we would encourage contacting Orkun Akin for direct assistance if you are interested in using this software for your research.

The Matlab scripts have been written and tested in version R2011a 64-bit for Mac and call functions from the Image Processing, Statistics, and Parallel Processing toolboxes. Some of the FEX sourced software (e.g. nonrigid\_version19) require compiling accompanying C files prior to running.

**Image Processing Workflow**

For a given data set, calls for the main suite of scripts are collected together with two Matlab structures, SIv and si, in an organizational file called s\_ProcessSeries\_Example.m.

**1. SIv:** Holds the series-specific variables.

**2. s\_buildSIstruct\_v1:** The first call of this script creates a folder structure that will house the raw and derived data. Later calls do not alter the existing folders but simply generates ‘si’, a Matlab structure with references to each sub-folder. Once the data folder is created, move the raw stacks into subfolder dated\_folder/one/grn/rawG/. Note that s\_buildSIstruct\_v1 can build a folder structure for two different data sets (i.e. odd and even) with two channels each (i.e. red and green). A single data set with one channel images will default to being called green (i.e. ‘grn’) and the some of the output images will be tagged with the suffix ‘G’.

**3. s\_getTimeStamps:** Prompts the user to select a folder with the data series. Make sure that the version of the data series used retains the original image capture time stamp. The output is saved in dated\_folder/one/suite\_output/txt\_files/ .

**4. s\_devTime:** Uses SIv.stagingDate and SIv.stagingMode to calculate the developmental time at the time of imaging. Output, ‘hrsAPF.txt’, is saved to ../txt\_files/ .

**5. s\_StkFileNames\_v2:** Writes a text file list of the stack names to ../txt\_files/ .

**6.** Run the FIJI macro FIJI\_sca.ijm on the raw data series. This step scales the z dimension of the stacks to produce voxels with 1:1:1 aspect ratio. When prompted for the scaled stack directory, choose dated\_folder/one/grn/scaG/. Note that the ‘duplicate’ call in this macro is an opportunity reduce the size of the scaled stacks, provided that the relevant image features remain within a smaller range of slices than the full number throughout the time series. The new range, if chosen, should be recorded in SIv.ScaZ\_stkLims.

**7. s\_TurnAngles\_v11:** This script calculates the rotation matrix that will generate a ‘top-down’ view of the medulla. Half-way through, it will wait for the user to ‘clean’ a background subtracted and morphologically opened version of the ‘anchor stack’, the image stack corresponding to the time point closest to 40 hAPF. This new stack is called ‘oStk.tif’ and is saved in ../img\_files/. Use FIJI to remove traces of the lamina and other non-R8 features and save the cleaned stack as ‘oStk\_fx.tif’ in the same ../img\_files/ folder. The principal output of the script, the transformation matrix, is called 'FijiXform.txt’ and it is saved in ../txt\_files/. FIJI plug-in TransformJ can be used to apply this matrix to the anchor stack for validation.

**8. s\_SliceRegister\_v5:** Registers a ROI from the center of the medulla, as calculated in s\_TurnAngles\_v11, through the time series. Output ../img\_files/RegSlices.tif can be viewed for quality control. Coordinates of the registered ROI are saved in ../txt\_files/RegArray.wk1.

**9. s\_GlobalAlign\_v6:** Computes transformation matrices to bring all time points into register with the top-down view of the anchor stack. Matrices saved in ../suite\_output/mat\_files/mFiji\_Global/ .

**10.** Run the FIJI macro FIJI\_sca.ijm on the scaled data series. When prompted for the registered stack directory, choose dated\_folder/one/grn/regG/. To ensure that the transformed stacks are all the same size, they are first embedded, centered, in a larger stack, then transformed, and finally cut to size. The limits of this final shaving step—the ‘crop’ call in the macro—are determined empirically by applying the transformations to three different stacks that sample the time series before the macro is run on the full set. The limits are chosen to retain most of the relevant R8 features while minimizing dark volume.

**11. s\_ShieldMask\_v2, s\_Shield2Shell\_v1, s\_ShellzMIP\_v3:** These three scripts should need minimal supervision. They generate a 3D mask that aims to capture R8 growth cones through the globally registered time series and use this mask to produce a top-down MIP of the R8 array through time. The final output is ../img\_files/ShellzMIP\_val-G.tif.

**12. s\_DefineGCs\_v4:** This script segments a masked projection of the ‘seed stack’, corresponding to the time point closest to 45 hAPF, and saves the result as 'seedMskMip.tif' in ../img\_files. The user is expected to clean-up this image, breaking up incorrectly joined growth cones (GCs) and removing spurious ROIs, and save the new curated version as 'seedMskMip\_fx.tif'.

**13. s\_TrackGCs\_v2:** Tracks the centroids of the defined GCs through the globally registered time series. Image outputs 'TrackGCMap.tif' and 'TrackGCMontage.tif' in ../img\_files/ can be viewed for quality control.

**14. s\_AlignGCs\_v7:** Registers each GC to itself in 3D through the time series. Saves transformation matrices for each GC at each time point in newly created sub-folders in ..suite\_output/mat\_files/mFIJI\_LocalG/. The center coordinate of the ROIs containing each GC are also recorded in these matrices, in positions (4,1),(4,2), and (4,3) for x, y, and z, respectively. The principal image output is ../img\_files/AlignGCZMIP2-G.tif. May take several hours to run, depending on the number of time points and GCs.

**15. s\_PlotGCExtension\_v10:** Tracks the tip position for each GC through the series. The principal image output is ../img\_files/tipTrak-G.tif. The fidelity of the tracking depends on image quality; output may require proof-reading and correction.