nature methods

Resource

https://doi.org/10.1038/s41592-023-01848-5

BigNeuron: a resource to benchmark and predict performance of algorithms for automated tracing of neurons in light microscopy datasets

In the format provided by the authors and unedited

Image quality measurement

To quantify image quality, we implemented a plugin in Vaa3D (RRID:SCR_002609, version 3.497, http://vaa3d.org) that computes the features described in detail in[1]. The following features are measured by the plugin.

Blur

- Focus score: a measure of the intensity variance across the image.

Saturation

- Percent maximal: percent of voxels at the maximum intensity value of the image.
- Percent minimal: percent of voxels at the minimum intensity value of the image.

Intensity

- Total intensity: sum of all voxel intensity values.
- Mean intensity, median intensity: mean and median of voxel intensity values.
- Std. intensity, MAD intensity: standard deviation and median absolute deviation (MAD) of voxel intensity values.
- Min. intensity, Max. intensity: minimum and maximum of voxel intensity values.

Threshold

- Otsu threshold: an automatically calculated threshold for each image that maximizes the inter-class variance between background and foreground (Otsu thresholding method[2]).

We also implemented Signal-to-Noise Ratio (SNR, Eq.1) and Contrast-to-Noise Ratio (CNR, Eq.2) Ratio quantifications, defining the boundary between foreground and background in the image volumes both as the average intensity and the Otsu threshold, respectively.

$$SNR = \frac{\overline{I_f}}{sd(I_b)} \quad \text{Eq. 1}$$

$$CNR = \frac{|max(I) - \overline{I}|}{sd(I_b)} \quad \text{Eq. 2}$$

Where I_f and I_b stand for the foreground and background intensity, respectively; I stands for the image intensity; max for the maximum value; and sd for the standard deviation.

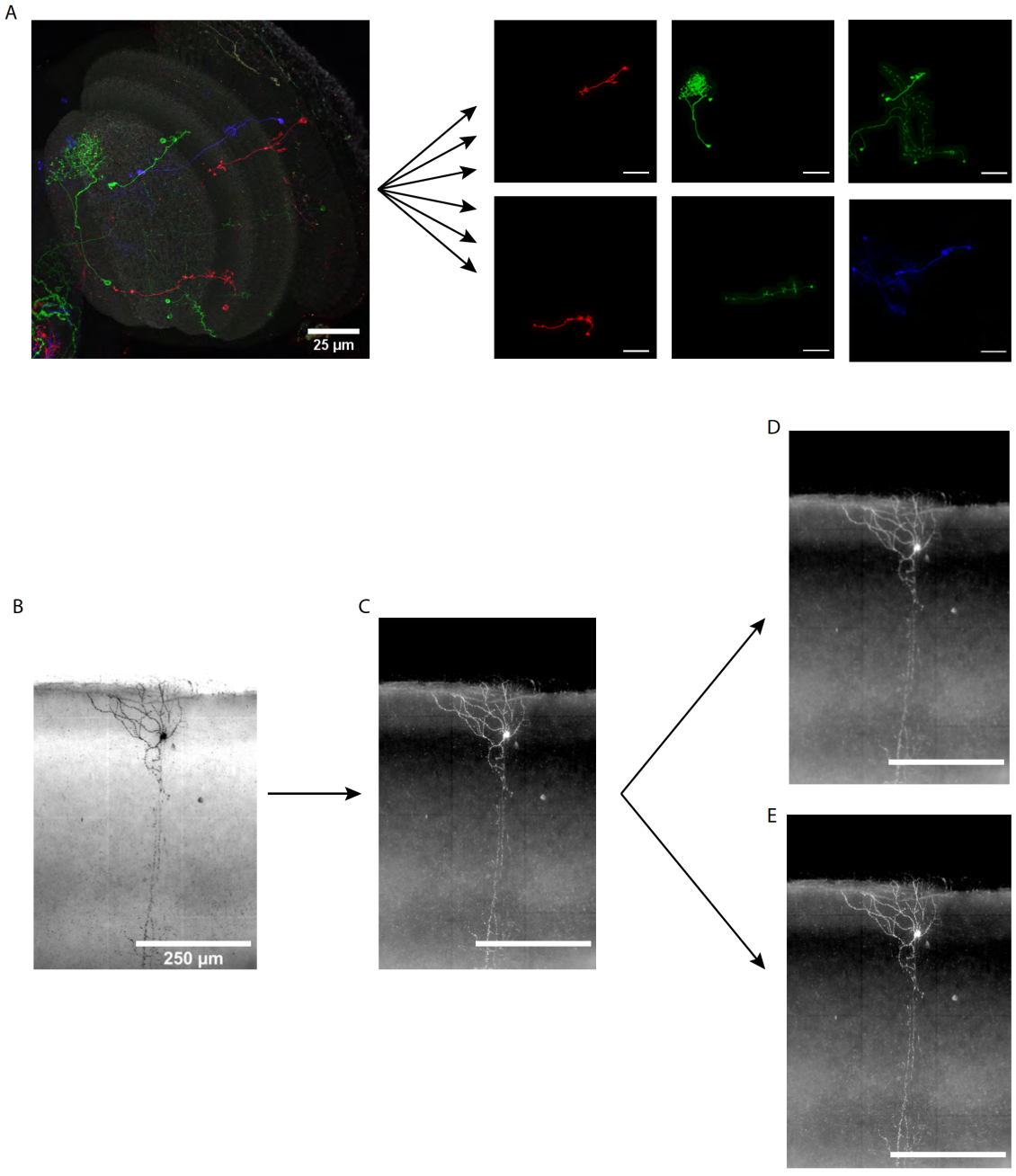
Image preprocessing

Due to varying properties of image acquisition pipelines among institutions, it is challenging to define a universal data preprocessing protocol for reconstruction. We performed the following preprocessing steps:

- MultiColor FlpOut (MCFO) color separation (Supplementary Figure 1): For MCFO data, we applied the "neuron_color_separator" plugin of Vaa3D to extract single neurons and save each of them into a separate file for reconstruction. When more than one neuron had the same color, we separated them with the connected component algorithm. The threshold is picked based on the voxel intensity histogram of the 3D volume. We start the search of connected components from voxels with intensity higher than 100 and take pixels with intensity lower than 10 as background to stop marching. Finally, we visually inspected all color separation results and filtered out cases in which more than one neuron were connected (i.e. had overlapping signals).
- Conversion to 8-bit single-channel images. We first simplified multiple color channel images to the main single channel. Subsequently, if the image had a 16-bit dynamic range, we converted it to 8-bit using linear rescaling. It is reasonable to argue that higher dynamic range could provide an advantage for automated reconstruction methods; however, the SNR is more relevant than adding precision in the dynamic range. The dynamic range of 8-bit images already allows SNRs that are sufficient for the tracing algorithms to work optimally. Going beyond those when the labeling and imaging quality is close to perfect may improve marginally their performance, but increasing the bit depth of images with high noise and low signal levels does not increase their SNR.
- Color inversion: We inverted the intensity of bright-field images for reconstruction

For FlyCircuit and FlyLight datasets, bench testing automatic reconstructions were obtained both for the raw datasets, and after preprocessing with one of the two following possibilities:

- Gaussian smoothing: If the raw image had clear noise in the signal along the neurites, we used Gaussian smoothing with 7x7x5 Gaussian kernels. To do so, we ran the "image_filters/Gaussian_Filter" plugin in Vaa3D.
- Anisotropic filtering: It is able to enhance the signal along neurites while suppressing noise. This is useful for filling gaps when the intensity is uneven spatially. We did this using the "image_filter/anisotropic_filters" plugin in Vaa3D.



Supplementary Figure 1. Preprocessing steps used before automatic reconstruction algorithm bench testing.

A Schematic view of an example of color separation for a brainbow dataset. The image on the left shows a 2D maximum intensity projection of a brainbow RGB dataset. The six images on the right show

separated individual neurons after using the “neuron_color_separator” plugin. The neurons labeled in the image are from the drosophila optic lobe from the Janelia Flylight dataset. **B** 2D maximum intensity projection of a brightfield microscopy dataset. Scale bar: 25 µm. **C** Shows the brightfield dataset after performing a color inversion. **D** Shows the inverted brightfield dataset after performing Gaussian smoothing. **E** Shows the inverted brightfield dataset after performing anisotropic filtering. The neuron in sections B-E is a pyramidal mouse neuron from barrel cortex provided by U.Goettingen. Scale bar: 250 µm.

Generation of gold standard reconstructions

We generated three files for each data entry: one 3D image stack file (in Vaa3D’s .v3draw or .v3dpbd format), one neuron reconstruction file (in the .swc format), and one linker file (in Vaa3D’s .ano format). Users can open any data entry conveniently by dragging and dropping the linker file (.ano) into the main window of Vaa3D. Since the original reconstructions validated by human annotators may not have consistent tags used to annotate dendritic, axonal, and somatic compartments, to avoid confusion and for simplicity, we uniformized all files so as include only two fixed node tags, 1 for the root and 3 otherwise.

Development of automatic reconstruction algorithms

All neuron tracing software is available from the latest binary release of Vaa3D. The reader is referred to the Vaa3D Github project page for more information (we documented the source code locations in https://github.com/BigNeuron/BigNeuron-Wiki/wiki/Neuron-Reconstruction-Algorithms). The tracing methods are available in the "main menu -> Plugins -> Neuron Tracing menu". Users can run these methods by using both GUI and the command line. Due to the limitation of dependency libraries for various tracing algorithms, the most complete set of tracing methods are available only for Linux.

Interactive data analysis app

Organization of the web-app

The web app has a set of panels that allow the users to select interactively data subsets based on their metadata, and metrics of interest. One left-sided panel allows users to choose the metrics of interest for dimensionality reduction and cluster analysis, and to choose which grouping variable is used for coloring in dimensionality reduction. A set of three bottom panels allows filtering the analyzed datasets by multiple selections of algorithms and datasets, and by filtering the range of values of a given metric. An additional bottom panel allows algorithm developers to upload a set of reconstructions obtained by self-developed algorithms to compare and benchmark algorithms developed in the future with the ones analyzed at the time of this publication.

The great amount of diversity in light microscopy image quality among datasets obtained by different teams hinders the advancement of the field by posing extra difficulties when using existing automated tracing algorithms. With dimensionality reduction and Clustering analysis tools, we aim to provide quantitative context on such diversity to study its impact on accuracy and identifying the algorithms that perform best in specific groups of images.

Dimensionality reduction

To identify the metrics that better separate different groups of reconstructions, we performed a Scaled Principal Component Analysis (PCA[3]). We analyzed features associated with each reconstruction (morphological properties and quality metrics) and the datasets they belong to (image quality features). To obtain the results of Fig. 3, only gold standard reconstructions were analyzed, including the default set of morphology and image quality features defined in the Shiny app and summarized in Supplementary Table 3. Before dimensionality reduction, the features with skewness greater than unity were log-transformed. We inverted negatively skewed distributions. This preprocessing ensures an approximately normal distribution of the input features[4]. PCA was performed using prcomp in R (version 4.0.3). We generated biplots in R (version 3.4.1) using ggbiplot (version 0.55) to reduce the number of variables into a smaller number of principal components that account for most of the variance. Barplots of the contributions of each variable to the two first principal components were obtained using the factoextra package (version 1.0.7). To explore possible nonlinear relationships between datasets, we obtained t-distributed stochastic neighbor embedding (t-SNE[5]) visualizations using the Rtsne package (version 0.15) with 500 maximum iterations and default parameters otherwise. For comparisons between different organisms we measured Sholl profiles, tree volumes and branch densities using the nat package in R (version 1.8.16). The same pacage was used to plot neuronal trees in the Shiny app. To obtain the centripetal bias plot, we used the TREES Toolbox (version 2.0).

Cluster analysis

We obtained the pairwise Pearson correlation between every pair of features of all the neuron reconstructions and plotted them as a heatmap using heatmaply (version 1.1.0). We ordered the features in the heatmap based on hierarchical clustering and showed dendrograms on the top and right of the plot. We performed hierarchical clustering within the "heatmaply_cor" function, with the Ward.D2 method[6]. Similarly, we obtained dendrograms for hierarchical clustering of all the datasets using morphology metrics, image quality metrics, and the combination of both. To allow users to focus on specific groups of datasets, we defined clusters based on Gaussian Mixture model-based clustering results using the "Mclust" function of the mclust package (version 5.4.7). We obtained the dendrogram of figure 3I using the "hc" function of mclust. We calculated coefficients of determination and p values with the "stat_cor" function of the ggpubr package (version 0.4.0).

Image quality metrics correlate with the accuracy of automated tracing

To explore how reconstruction quality varies among different images, we tested how the distance between consensus reconstructions and gold standard trees correlate with every feature of the datasets. We performed this analysis by obtaining a correlation matrix among reconstruction quality, image quality, and neuron morphology features of the consensus reconstruction results (Extended Data Fig. 5A). Distance metrics correlated with diverse image quality and tree morphology features: CNR (Otsu) in SWC nodes, the median intensity of the image, parent-daughter ratio, and remote bifurcation angle (Extended Data Fig. 5A, bottom left cluster). A second cluster of features showed a negative correlation with distance metrics: focus score (together with its equivalent obtained in SWC nodes), the percentage of image voxels with minimal intensity (together with its equivalent in SWC nodes), the average diameter of the automatic reconstruction segments and the standard deviation of the image intensity in SWC nodes (Extended Data Fig. 5A, top right cluster). Finally, a set of features showed little correlation with the quality of the consensus reconstructions: the number of tips, maximum branch order, total length, maximum path distance (Extended Data Fig. 5A, center cluster). Thus, as a representative example, the percent of different structure between consensus reconstructions and gold standards inversely correlates with the Focus score in SWC nodes (Extended Data Fig. 5B, Pearson's correlation coefficient: R^2=0.13, p=4.1e-6). By contrast, the percent of different structures positively correlates with the parent-daughter ratio (average ratio of the number of child nodes for each parent node in the tree structure; Extended Data Fig. 5C, R^2=0.23, p=1.5e-10) and also negatively correlates with the average remote bifurcation angle (Extended Data Fig. 5D, R^2=0.19, p=1e-8), indicating those tree morphology features have high values in low accuracy automatic reconstructions.

Brief description of major methos for automated neuron tracing

The diverse algorithms bench-tested in BigNeuron cover various major methods:

(1) All path-pruning methods include APP1, APP2 and Fast Marching Spanning tree all rely on the idea of defining nodes that oversample the foreground space in the image, and those are pruned iteratively to obtain a succinct tracing.

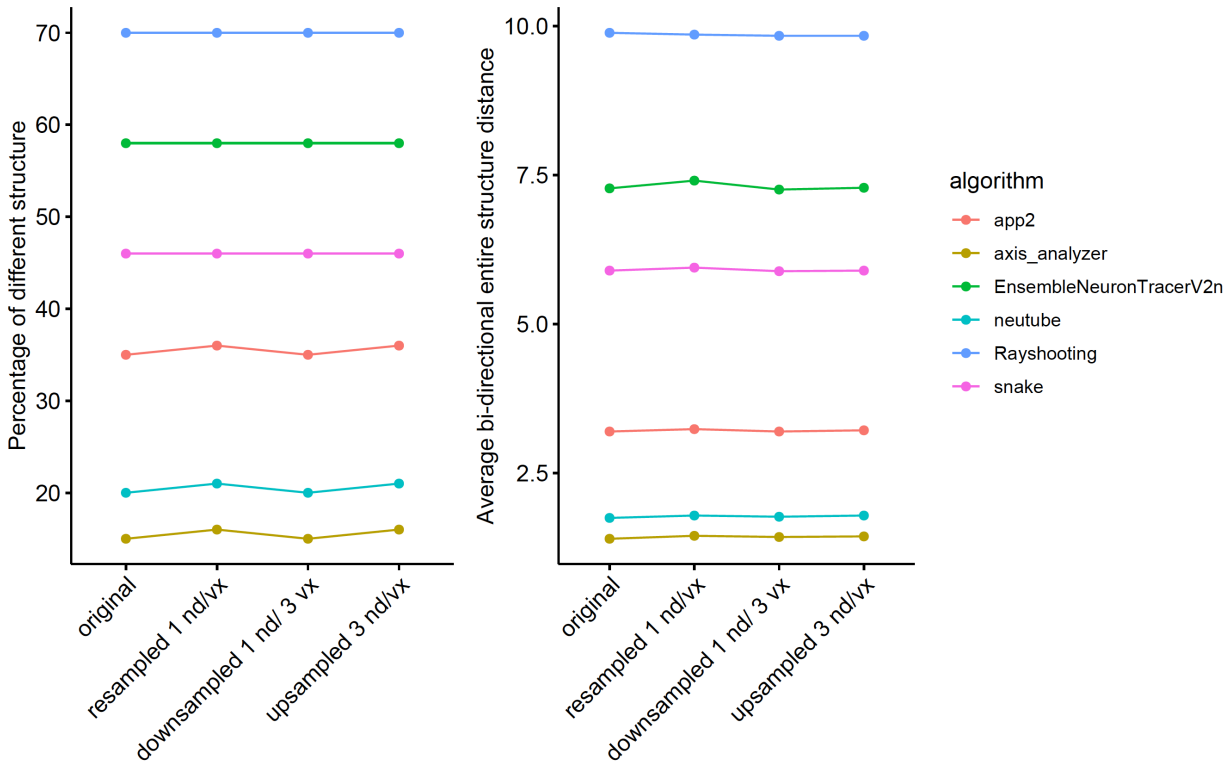
(2) The Ensemble Neuron Tracer variants, as mentioned in the main text, rely on a single base tracer to generate tracing segments from perturbations of the input data. Fusing them in an as long as possible single tracing result.

(3) The SimpleTracing variants rely on a distance-field transformation of the image data to explore the foreground signal from an automatically determined seed point.

(4) FarSight Snake and NeuTube rely on identifying tubular structures in the image data and generating trees from the obtained segmentations.

(5) AxisAnalyzer, MeanShift and NeuTu autotrace rely on methods for binarizing the original image and use different strategies to connect the nodes defined by positive voxels.

(6) LCMboost and SmartTracing use initial results from base tracers and then complete them using machine-learning frameworks to identify image areas that contain neuron signal.



Supplementary Figure 2. Tracing accuracy metrics are robust with node density variation.

The line plots show the values for tracing accuracy metrics (percentage of different structure in the left panel and average bi-directional entire structure distance in the right panel). The colors indicate values obtained for different automated tracing algorithms in comparison to gold standard trees. We measured

accuracy in the original files and after resampling them to have: (1) 1 node per voxel, (2) downsampled to have 1 node every 3 voxels and (3) upsampled to have 3 nodes for every voxel. These results were obtained with image with id=99 of the Gold166 dataset

Supplementary Note – BigNeuron Shiny app license

The basic modules and standard plugins of the BigNeuron Shiny app use the slightly revised MIT License. Note, however, that other modules and extensions may use different licenses as long as such licenses are compatible with this license.

References

1. Bray, M.-A. & Carpenter, A. E. Quality Control for High-Throughput Imaging Experiments Using Machine Learning in Cellprofiler. *High Content Screening: A Powerful Approach to Systems Cell Biology and Phenotypic Drug Discovery* (eds. Johnston, P. A. & Trask, O. J.) 89–112 (Springer, 2018). doi:10.1007/978-1-4939-7357-6_7.
2. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979).
3. Wold, S., Esbensen, K. & Geladi, P. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **2**, 37–52 (1987).
4. Polavaram, S., Gillette, T. A., Parekh, R. & Ascoli, G. A. Statistical analysis and data mining of digital reconstructions of dendritic morphologies. *Front. Neuroanat.* **8**, 138 (2014).
5. van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
6. Murtagh, F. & Legendre, P. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *J. Classification* **31**, 274–295 (2014).

APPENDICES

Open Data Agreement

The *BigNeuron* project imposes little restriction on the contribution, use, publication and distribution of the data, while providing as much flexibility as possible for contributors to continue their own work.

BigNeuron calls for the contribution of neuronal images for public domain bench testing. For any contributed neuron images, a set of reconstructions will be produced using the ported automated neuron tracing methods. The data contributors can use such reconstructions freely given the *BigNeuron* project is appropriately cited.

Contributors of such neuron images of *BigNeuron* project may use, copy, distribute, publicly perform, publicly display or create derivative works of the neuron reconstructions corresponding to respective images for research, noncommercial, and commercial purposes, given that *BigNeuron* project as well as the respective reconstructions and algorithms or implementations used in *BigNeuron* are appropriately referenced. In addition, where the reconstructions contain links to downloadable software applications, services or tools, image contributors may download and use such applications, services or tools as long as such contributors adhere to any license terms and conditions provided with those applications, services or tools.

Image contributors may not post reconstructions on social media or other third-party websites that require image contributors to acknowledge that they own the data they post (e.g., YouTube, Flickr and Twitter). Image contributors agree that they will not use the *BigNeuron* data in any manner that would violate anyone else's rights, such as copyright, trademark, patent, privacy or other rights. This includes removing any copyright, trademark or other proprietary notices from the reconstructions. Image contributors may not create hyperlinks to the *BigNeuron* resources (e.g. website, database, documentation site) that portray this project in a false or misleading light. Image contributors agree that they will only make lawful use of the *BigNeuron* Data and Resources in compliance with all federal, state, and local laws and regulations.

Image contributors may, and are encouraged to, develop new methods, applications, interfaces or other inventions or works that improve the use of, and build upon, the reconstructions. In order to make the reconstructions available to image contributors and others, however, the *BigNeuron* project must preserve its freedom to innovate. If image contributors develop improvements based on or utilizing the reconstructions, and such image contributors obtain any proprietary rights in or to that improvement, the image contributors and their successors or assigns agree that image contributors will not assert any claim for infringement against the *BigNeuron* project for the use of any improvement that was independently developed by or on behalf of the *BigNeuron* project. Additionally, the *BigNeuron* project retains its rights, title and interest in any reconstructions that are part of or are used by image contributors to create an Improvement.