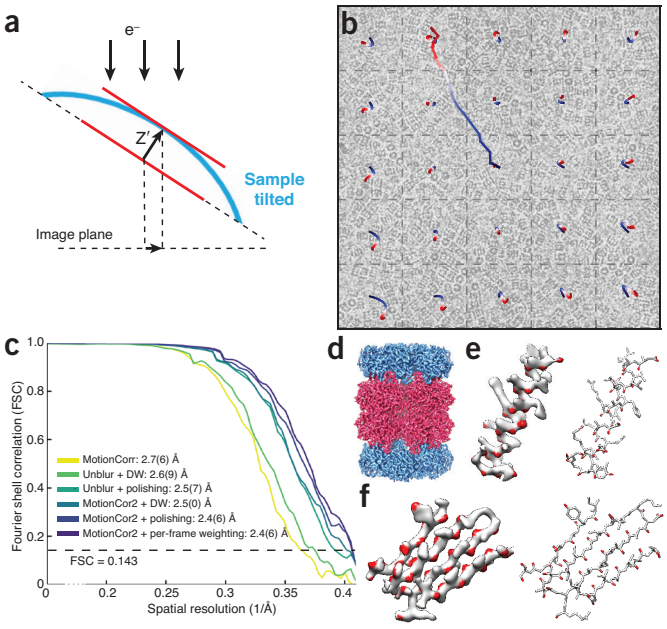


Figure 1 | Local motion correction by MotionCor2. (**a**) Schematic drawing illustrates that when the sample is tilted the observable motion in the image plan is the projection of *z*-motion produced by doming of the sample under electron beam. (**b**) Image of frozen hydrated archaeal 20S proteasome overlaid with the traces of global motion based upon whole-frame alignment (long trace originated from the center of image) and each patch predicted from the polynomial function. (**c**) Fourier shell correlation (FSC) curves of 3D reconstructions determined using micrographs corrected by Unblur with dose weighting, Unblur followed by particle polishing, correction by MotionCor2 with dose weighting, MotionCor2 followed by particle polishing and MotionCor2 with per-frame B-factor weighting. (**d**) 3D reconstruction of archaeal 20S proteasome filtered to 2.5-Å resolution and sharpened by a temperature factor of −103.8 Å^2. (**e**) Density of an α helix from the map, with resolved oxygen atom functional groups colored in red. Visualization of main chain carbonyls requires resolution below 3 Å. The refined atomic model is shown side by side for comparison. (**f**) As in **e**, but showing a β sheet.

computationally intensive particle polishing in RELION can be skipped. Importantly, it also works on a wide range of data sets, including cryo tomographic tilt series.

Data availability statement. The refined coordinates of archaeal 20S proteasome and the density maps of both archaeal 20S proteasome and TRPV1 are included as **Supplementary Data**.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper (doi:10.1038/nmeth.4193).

ACKNOWLEDGMENTS

We thank X. Li for helpful discussion during the initial stage of this work. We also thank M. Braunfeld for supporting the cryo-EM facility at UCSF, G. Greenan (Department of Biochemistry and Biophysics, University of California San Francisco) for providing his cryo-tomographic tilt series collected on a *Drosophila* centriole, and C. Kennedy for supporting the computational infrastructure for processing cryo-EM data. This work was supported in part by grants from National Institute of Health—R01GM031627 to D.A.A. and P01GM111126, P50GM082250, R01GM082893 and R01GM098672 to Y.C. Y.C. and D.A.A. are Investigators of Howard Hughes Medical Institute.

AUTHOR CONTRIBUTIONS

S.Q.Z. implemented the algorithm and wrote all codes for MotionCor2. D.A.A. contributed to algorithm development. S.Q.Z, E.P., Y.C. and D.A.A. designed experiments to evaluate the performance of MotionCor2. K.A.V. designed initial experiments for camera defect correction. S.Q.Z. and E.P. carried out image processing. E.P. and J.-P.A. collected low-defocus cryo-EM images of 20S proteasomes. S.Q.Z., E.P., Y.C. and D.A.A. wrote the manuscript. All authors participated in discussion and revision of the manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Shawn Q Zheng[1,2], Eugene Palovcak[1], Jean-Paul Armache[1], Kliment A Verba[1], Yifan Cheng[1,2] & David A Agard[1,2]

[1]Department of Biochemistry and Biophysics, University of California, San Francisco, San Francisco, California, USA. [2]Howard Hughes Medical Institute, University of California, San Francisco, San Francisco, California, USA.
e-mail: ycheng@ucsf.edu or agard@msg.ucsf.edu.

Published online 27 February 2017; doi:10.1038/nmeth.4193

1. Cheng, Y. *Cell* **161**, 450–457 (2015).
2. Kühlbrandt, W. *Science* **343**, 1443–1444 (2014).
3. Brilot, A.F. *et al. J. Struct. Biol.* **177**, 630–637 (2012).
4. Li, X. *et al. Nat. Methods* **10**, 584–590 (2013).
5. Liao, M., Cao, E., Julius, D. & Cheng, Y. *Nature* **504**, 107–112 (2013).
6. Bai, X.C., Fernandez, I.S., McMullan, G. & Scheres, S.H. *eLife* **2**, e00461 (2013).
7. Rubinstein, J.L. & Brubaker, M.A. *J. Struct. Biol.* **192**, 188–195 (2015).
8. Grant, T. & Grigorieff, N. *eLife* **4**, e06980 (2015).

Automatic tracing of ultra-volumes of neuronal images

To the Editor: Despite substantial advancement in the automatic tracing of neuronal morphology in recent years[1–3], it is challenging to apply the existing algorithms to large image data sets containing billions or even trillions of voxels. Most neuron-tracing methods published to date were not designed to handle such data. We introduce UltraTracer (**Fig. 1**), a solution designed to extend any base neuron-tracing algorithm to allow the tracing of ever-growing data volumes. We applied this approach to neuron-tracing algorithms with different design principles and tested it on human and mouse neuron data sets that have hundreds of billions of voxels. Results indicate that UltraTracer is scalable, accurate, and more efficient than other state-of-the-art approaches.

The core algorithm of UltraTracer (**Fig. 1**) reconstructs a neuron structure from the available image data on the basis of a formulation of maximum-likelihood estimation. The underlying assumption is that the occurrence of a specific neuron structure could be modeled using the joint probability of all of its subparts given the image. Briefly, UltraTracer iteratively factorizes the joint probability based on progressive maximization of conditional probabilities of the occurrence of salient and continuous subparts of a neuron (**Supplementary Note**). Therefore, UltraTracer explores an image by following where the neurite signal goes, on the basis of either adaptive windows generated based on the already reconstructed neuron structure or certain domain knowledge (prior information or statistics) of neuron morphology, to help refine the choice of the next tracing subarea (**Supplementary Note**). This process repeats until the neuron structure grows as completely as possible. We designed the UltraTracer software to quickly extract an arbitrary subvolume of interest from large neuron image files (**Supplementary Note**), and thus smoothly traced an image archive without the need to load a large number of image voxels into computer memory.

As a crucial utility that was not previously available to reconstruct large-scale data sets, UltraTracer extends an arbitrary base tracer to make it possible to trace an ever-increasing image volume. We tested this by considering ten representative base tracing algorithms ported to BigNeuron[3] (https://github.com/BigNeuron/BigNeuron-Wiki/wiki/Neuron-Reconstruction-Algorithms) that have different design principles, performances, and output formats (**Supplementary Figs. 1**, **2**, and **3**; **Supplementary Note**). The performance gain of UltraTracer over the direct use of certain base tracers was within the range of 3–6 times (**Supplementary Fig. 1b**). UltraTracer results were accurate, as their average spatial distances to independent manual reconstructions were around 3 voxels, comparable to the spatial distance of the manual reconstructions themselves (3.56 voxels) (**Supplementary Fig. 1b**). In addition, for two base tracers, NeuTube[4] and MOST[5], UltraTracer had a gain of 10–30-fold in tracing accuracy (**Supplementary Fig. 1b**). Testing of six other base tracers (**Supplementary Fig. 2**) indicated similar improvement. When a computer with smaller memory was used or the image volume increased greatly, UltraTracer was consistently superior to the conventional approach (**Supplementary Fig. 3**).

The APP2 algorithm[6] was a good base tracer, in terms of speed–accuracy trade-off (**Supplementary Fig. 1**), for both laser-scanning and brightfield images (**Supplementary Figs. 3**–**7**). The APP2-based UltraTracer scaled robustly in tracing the sparse neuronal structures in images with 521 billion voxels, reducing the data volume in tracing between 3 and 40 times (**Supplementary Fig. 3**). Typically a bigger data-volume reduction rate was achieved for a larger image volume. Measured in terms of spatial distance, bifurcation points, and five other morphological and topological features, and compared against the statistics drawn from collections of reconstructions produced using control images (**Supplementary Note**), the accuracy of reconstructions produced by UltraTracer was consistent with that of reconstructions generated using the traditional approach when the image data set could be accommodated by the computer memory (**Supplementary Fig. 3**, bottom left).

We used UltraTracer to combine multiple different base tracers (**Supplementary Note**; **Supplementary Figs. 8** and **9**), for example using APP2 in the soma area while using NeuTube and MOST to trace curvilinear structures. In a more complicated case, for every adaptively searched image region, we profiled the reconstructions generated by several base tracers, and then chose either the best reconstruction or their consensus as the result from the current image region (**Supplementary Fig. 9**). In this way UltraTracer could provide more consistent reconstructions compared to manual work. We also used UltraTracer to reconstruct human neurons, including their axons and dendrites, from separate but serial slices of brain tissue (**Supplementary Note**; **Supplementary Fig. 10**). Additional information about the algorithm can be seen in **Supplementary Figures 11**–**13**.

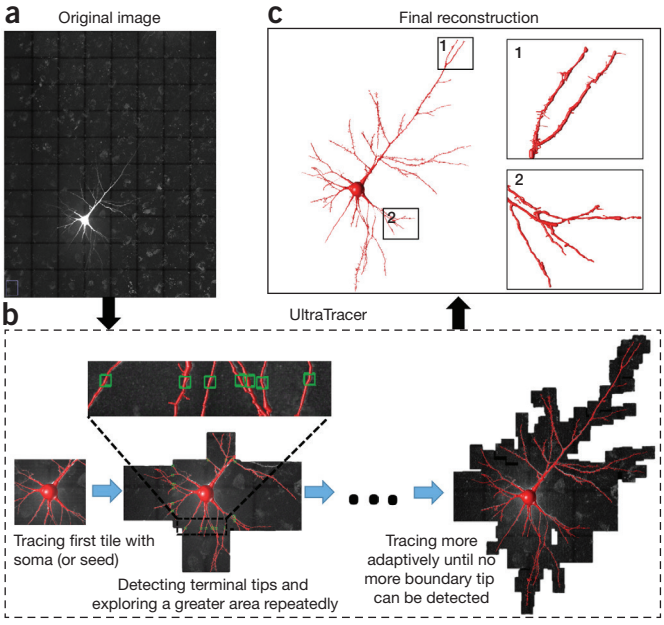


Figure 1 | Workflow of UltraTracer for tracing a large 3D image volume. (**a**) 3D confocal image stack of a Lucifer-Yellow-labeled human pyramidal neuron. The voxel size is 0.18 × 0.18 × 0.5 μm, and the overlaid grid (black lines) indicates how the image volume is subdivided into uniform tiles. (**b**) UltraTracer first traces the subarea containing the soma and then detects the neuron terminal tips in the reconstruction, and adaptively explores and traces neighboring subareas. Green boxes indicate terminal tips detected in tracing a subarea. (**c**) Final reconstruction produced by UltraTracer, with zooms of two parts for detailed visualization.

Data availability. UltraTracer is open source and available in Vaa3D software (vaa3d.org) and as **Supplementary Software**. The sample data are publicly available and can be downloaded from GitHub (https://github.com/Vaa3D/Vaa3D_Data/releases/download/v0.9/ultratracer_testing_data.zip).

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We thank the Allen Institute for Brain Science and data contributors to the BigNeuron project for providing neuron data sets. This work was funded by the Allen Institute for Brain Science. The authors wish to thank the Allen Institute founders, P.G. Allen and J. Allen, for their vision, encouragement and support.

AUTHOR CONTRIBUTIONS

H.P. conceived this project, designed and managed this study, proposed the theoretical framework of the method, and wrote the paper with assistance from Z.Z. and other coauthors. Z.Z. developed the tip-queue-based neuron growth algorithm, implemented the software, and generated results, with the help of coauthors.

COMPETING FINANCIAL INTERESTS

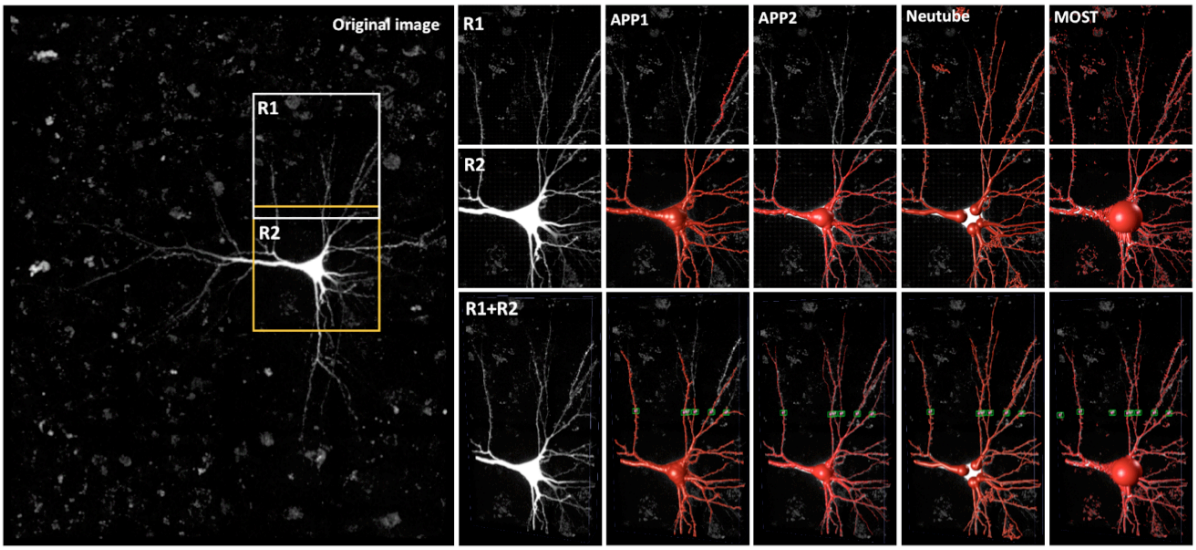
The authors declare no competing financial interests.

Hanchuan Peng[1,5], Zhi Zhou[1,5], Erik Meijering[2], Ting Zhao[3], Giorgio A Ascoli[4] & Michael Hawrylycz[1]

[1]Allen Institute for Brain Science, Seattle, Washington, USA. [2]Erasmus University Medical Center Rotterdam, Rotterdam, the Netherlands. [3]Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, Virginia, USA. [4]Krasnow Institute for Advanced Study, George Mason University, Fairfax, Virginia, USA. [5]These authors contributed equally to this work.
e-mail: hanchuanp@alleninstitute.org

1. Helmstaedter, M. *Nat. Methods* **10**, 501–507 (2013).
2. Acciai, L., Soda, P. & Iannello, G. *Neuroinformatics* **4**, 353–367 (2016).
3. Peng, H. *et al. Neuron* **87**, 252–256 (2015).
4. Zhao, T. *et al. Neuroinformatics* **9**, 247–261 (2011).
5. Wu, J. *et al. Neuroimage* **87**, 199–208 (2014).
6. Xiao, H. & Peng, H. *Bioinformatics* **29**, 1448–1454 (2013).

A



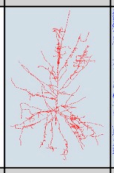
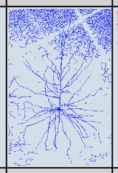
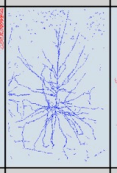
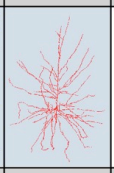
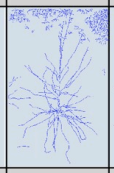
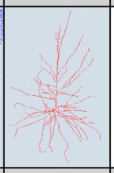
B

Algorithm	APP1		APP2		Neutube		MOST	
Tracing Method	TR	UT	TR	UT	TR	UT	TR	UT
Morphology Reconstruction								
BASDM (voxels) (against 2 ind. manual tracings)	1.83, 2.35	1.99, 2.50	2.30, 3.17	2.78, 3.57	179.72, 173.71	5.17, 5.42	39.83, 38.83	3.01, 3.42
Peak Memory (GB)	77.18	14.21	31.23	2.82	27.9	1.92	4.61	0.64
Tracing Time (s)	956.94	1132.82	69.95	115.67	1322.50	3317.86	22.43	29.65
Total Cost (PM x TT)	73856.63	16097.37	2184.54	326.19	36897.75	6370.29	103.40	18.98
Performance Gain (UT/TR)	3.59		5.70		4.79		4.45	

Supplementary Figure 1

UltraTracer extends and improves various base tracers to reconstruct large image volumes.

A. *UltraTracer* with four base tracers APP1, APP2, NeuTube, and MOST (Supplementary Note) applied to image regions R1 and R2. B. Comparison of *UltraTracer* and the direct use of base tracers. TR: traditional method (i.e. using a base tracer directly to reconstruct the entire 3-D image volume); UT: *UltraTracer*; BASDM: Best Average Spatial Distance compared with Manual reconstructions; PM: Peak computer-Memory; TT: Tracing Time. The image volume used has 2111×3403×291 voxels. Two independent human manual reconstructions were used for comparison; their BASD (Best Average Spatial Distance) is 3.56 voxels.

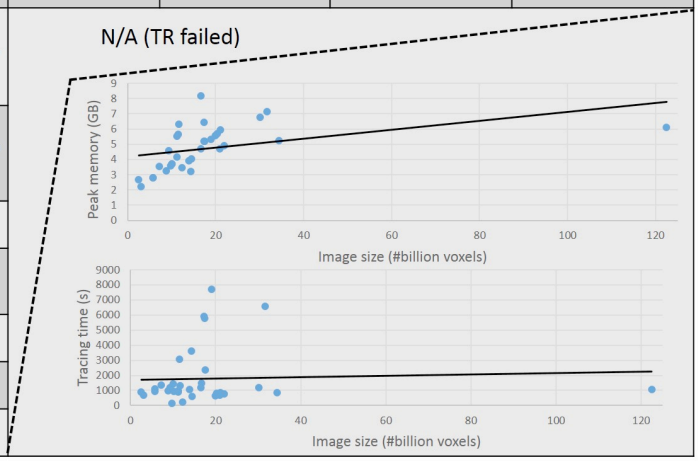
Algorithm	Snake		MST		NeuroGPSTree		Rivulet		TReMAP		nctuTW	
Tracing Method	TR	UT	TR	UT	TR	UT	TR	UT	TR	UT	TR	UT
Morphology Reconstruction	N/A						N/A				N/A	N/A
BASDM (voxels) (against 2 ind. manual tracings)	N/A	19.02, 19.08	228.23, 219.97	217.61, 210.52	69.17, 65.48	4.70, 4.72	N/A	3.98, 4.36	183.50, 177.83	1.67, 2.34	N/A	N/A
Peak Memory (GB)	N/A	34.18	19.46	5.12	18.18	4.61	N/A	10.62	4.99	3.71	N/A	N/A
Tracing Time (s)	N/A	3406.20	37324.05	1138.30	1966.21	2033.30	N/A	31646.52	758.46	1080.30	N/A	N/A
Total Cost (PM x TT)	N/A	116423.92	726326.01	5828.10	35745.70	9373.51	N/A	336086.04	3784.72	4007.91	N/A	N/A
Performance Gain (UT/TR)	N/A		124.62		3.81		N/A		0.94		N/A	

Supplementary Figure 2

Comparison of *UltraTracer* and the direct use of 6 additional base tracers on a human neuron image stack.

These 6 base tracers including Snake (Narayanaswamy, et al, 2011), Minimum Spanning Tree (MST as used by a number of groups independently; the original idea could be referred as Dijkstra, 1959), NeuroGPSTree (Quan, et al, 2016), Rivulet (Liu, et al, 2016), TReMAP (Zhou, et al, 2016), and nctuTW(Lee, et al, 2012). Two base tracers, Snake and Rivulet, were not able to generate the reconstruction using TR, since their usage of computer-memory exceeded the available memory of the testing computer (128 GB). One base tracer, nctuTW, failed to generate the reconstruction using TR or UT because it was too slow (in fact, it could not even produce a reconstruction for a 768×768×291 voxel sub-volume of the human neuron image stack within three hours). The image stack is the same one used in Supplementary Figure 1B.

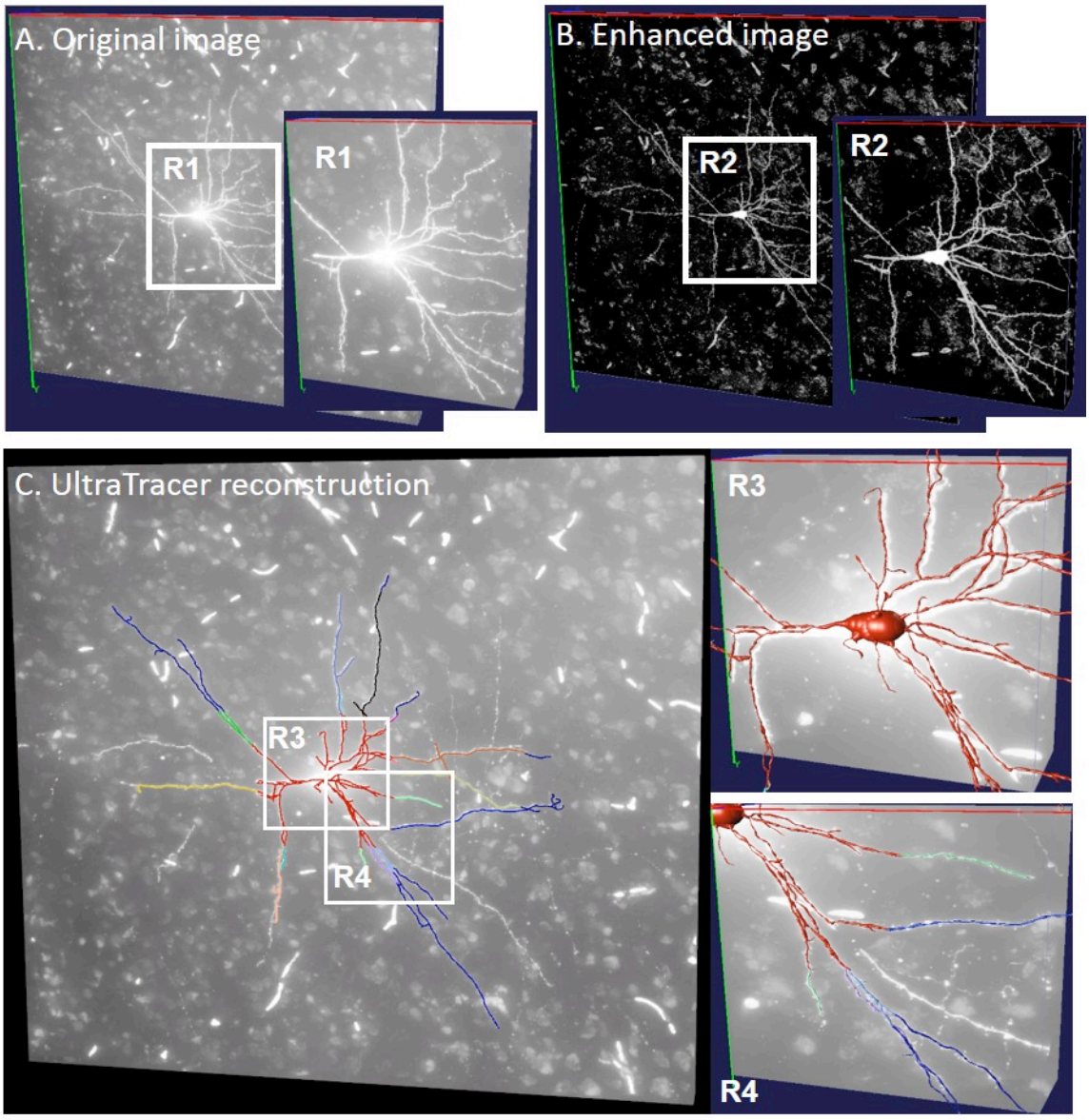
Neuron			1		2		3		4		5		6		7	
Bounding box dimension (voxels)			1639x1638x175		1637x1637x186		3899x3899x291		7469x9308x206		15223x12104x644		11271x11254x966		29880x18586x939	
Reconstruction approach			TR	UT	TR	UT	TR	UT	TR	UT	TR	UT	TR	UT	TR	UT
Traced volume (billion voxels)			0.470	0.098	0.498	0.117	4.423	1.611	14.321	3.006	118.662	6.291	122.531	6.979	521.471	13.310
Fold reduction in data volume			4.80		4.26		2.75		4.76		18.86		17.56		39.18	
Reconstruction consistency (UT and TR)	Average spatial distance, percentage of structure difference		**1.16 voxels** (2.06 +/- 0.49 voxels), **4.70%** (13.16% +/- 2.21%)		**2.11 voxels** (1.46 +/- 0.83 voxels), **3.78%** (10.07% +/- 1.91%)		**0.26 voxels** (0.94 +/- 0.22 voxels), **1.07%** (3.81% +/- 0.53%)		N/A (TR failed)							
	Average distance of matching bifurcation points, percentage of matched bifurcation pairs		**0.40 voxels** (1.46 +/- 0.43 voxels), **96.47%** (80.39% +/- 4.12%)		**0.01 voxels** (2.49 +/- 0.57 voxels), **93.92%** (67.61% +/- 4.57%)		**0.44 voxels** (2.56 +/- 0.64 voxels), **92.36%** (77.08% +/- 2.86%)									
	Global, local, topological features (relative difference)	Total length	**0.75%** (3.69% +/- 1.98%)		**2.04%** (1.76% +/- 1.37%)		**0.83%** (0.49% +/- 0.30%)									
		Total surface area	**0.46%** (4.22% +/- 2.93%)		**0.73%** (3.63% +/- 2.24%)		**0.77%** (1.12% +/- 0.85%)									
		Total volume	**0.08%** (6.89% +/- 4.82%)		**0.80%** (7.72% +/- 4.73%)		**1.97%** (2.47% +/- 1.88%)									
		Average diameter	**0.25%** (3.16% +/- 2.13%)		**1.80%** (2.84% +/- 1.46%)		**2.63%** (1.10% +/- 0.64%)									
		Hausdorff dimension	**1.42%** (1.62% +/- 1.07%)		**1.24%** (1.44% +/- 1.01%)		**0.15%** (1.46% +/- 1.95%)									



Supplementary Figure 3

UltraTracer (with base tracer APP2) is scalable with respect to ultra-volumes of neuron images, without compromising the tracing accuracy in terms of spatial distance, morphological and topological features.

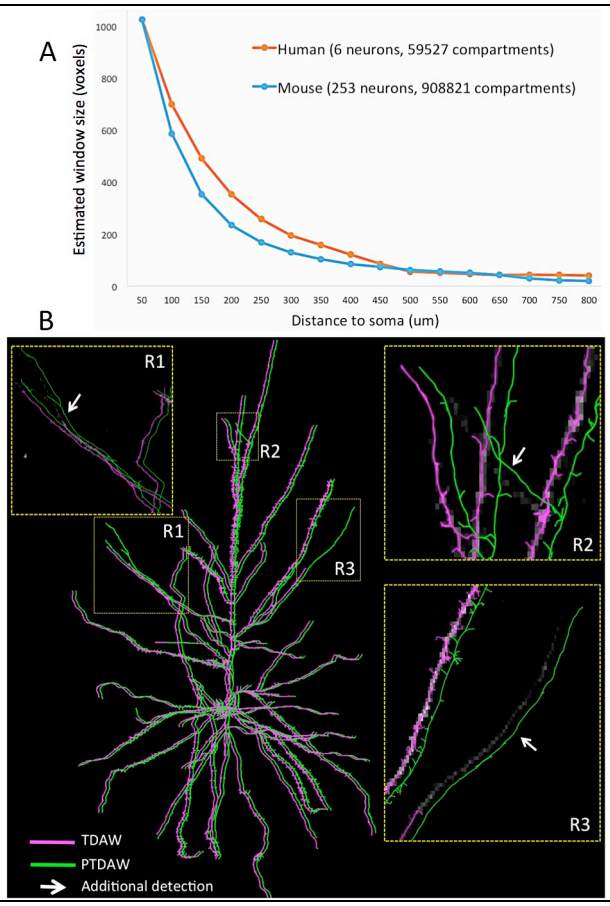
TR: Traditional approach. UT: *UltraTracer*. Testing data: neurons 1, 2, 3, and 4 are confocal image stacks of human pyramidal neurons, neurons 5 and 7 are confocal image stacks of mouse pyramidal neurons, neuron 6 is a brightfield image stack of human pyramidal neuron. In reconstruction-consistency testing of TR and UT based on various features, the “percentage of structure difference” of two reconstructions measures the portion of their visible difference (the nearest matching reconstruction nodes in two tracings are more than 2-voxel apart), the “percentage of matched bifurcation pairs” is defined as the portion of reciprocally best matching bifurcation points divided by the average number of bifurcation points of two reconstructions, the “total length”, “total surface”, and “total volume” are the length, the surface, and the volume of all neuronal compartments in reconstructions, the “average diameter” is the average diameters of all compartments in a reconstruction, the “Hausdorff dimension” (Falconer, 2004) measures the fractal dimension of reconstructions. In parentheses, the statistics (mean +/- s.d.) derived from TR-reconstructions using 59 rotated images (every 6 degrees around the center of XY-plane) for each neuron are shown as controls. Bottom-right inset: Regression analysis of peak memory and tracing time versus the image volume tested on 31 brightfield images.



Supplementary Figure 4

Application of *UltraTracer* to brightfield imaging image stacks of mouse V1 neurons.

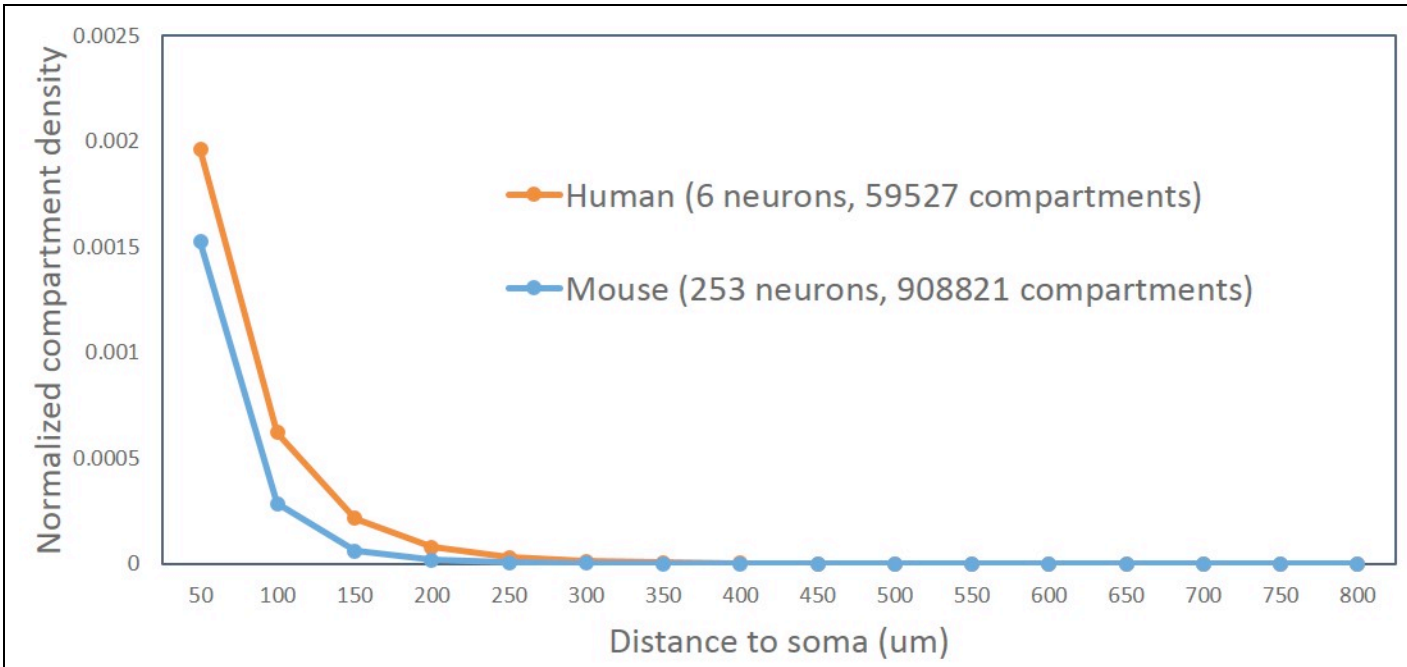
A. An example of brightfield image. B. Enhanced image using an adaptive approach (Zhou, et al, 2015). C. *UltraTracer* reconstruction based on the enhanced image in B. Different colors indicate reconstructions from different image regions.



Supplementary Figure 5

UltraTracer enhanced by incorporating prior knowledge of the adaptive subarea (window) size in tracing, which was learned from large-scale statistics of mammalian neuron reconstructions.

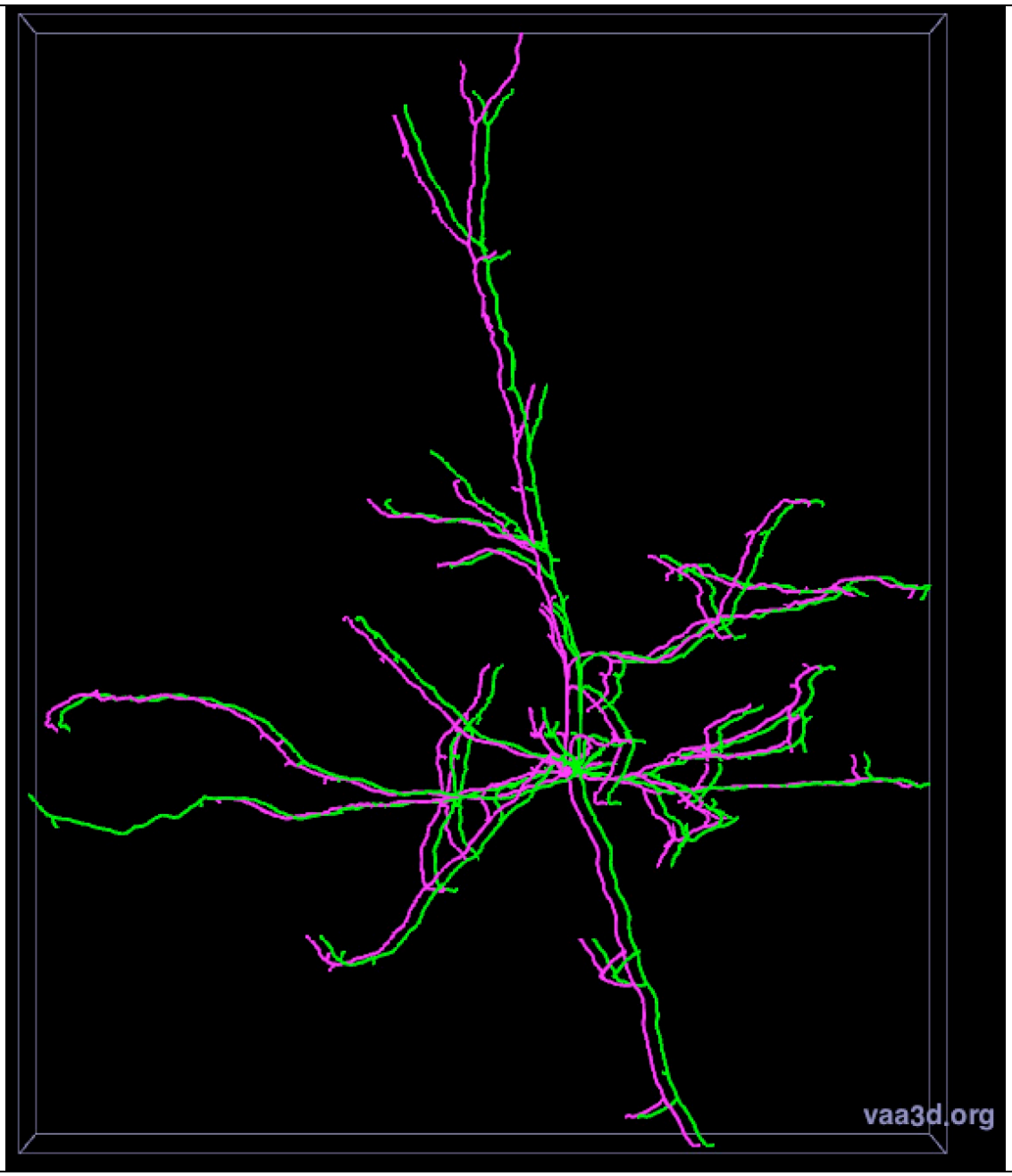
A. The estimated window size (in x, y, and z) as a function of the distance of a neuron compartment to the soma. The maximum window size was set to be 1024 voxels. B. Comparison results of two tracings, one with the TDAW method (magenta) and another with PTDAW (green), where the prior is the estimated window size in A. In each zoom-in region (R1 ~ R3), the gray-scale image voxels are also displayed. The two reconstructions are slightly offset in x-direction for better visualization.



Supplementary Figure 6

Average neuron-compartment density as a function of the distance between the neuron-compartment and the soma.

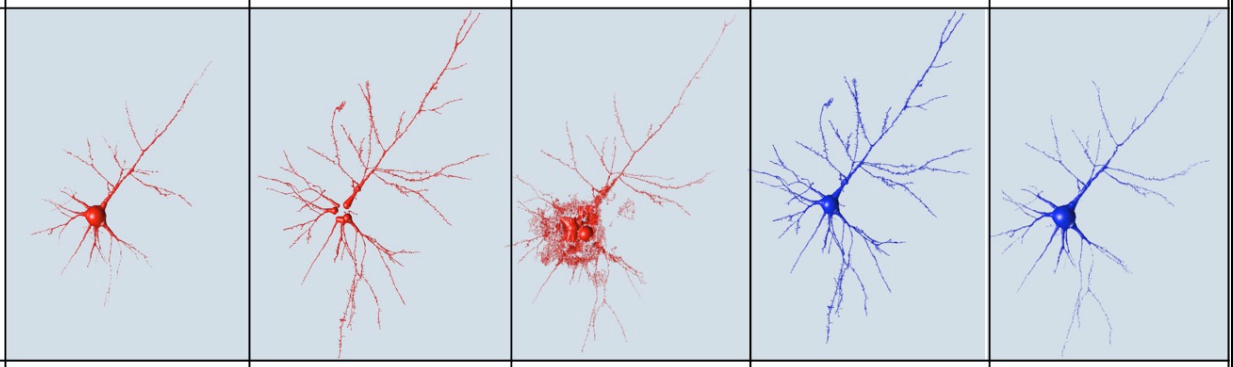
This information was used as a look-up table for PTDAW to avoid potential skewed estimation based on any extreme cases.



Supplementary Figure 7

Tracing results of TDAW (magenta) and PTDAW (green) for a mouse pyramidal cell (voxel size 0.143μm×0.143μm×0.28μm).

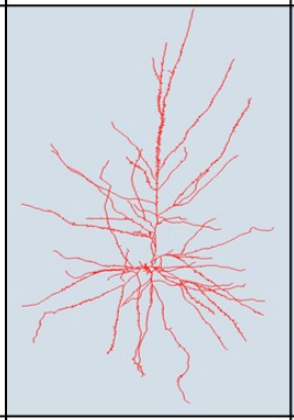
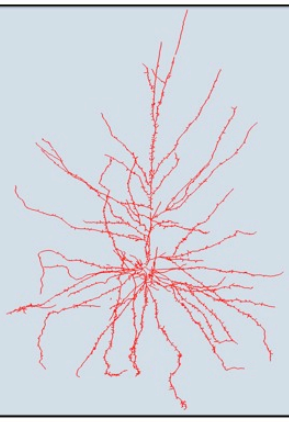
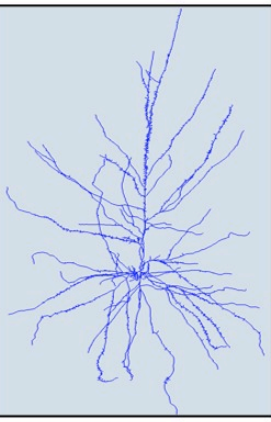
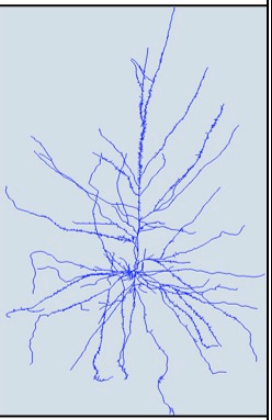
Reconstructions are slightly offset in x-direction for better visibility.

Algorithm	APP2	Neutube	MOST	APP2+Neutube	APP2+MOST
Morphology Reconstruction					
Total Scanned Areas (billion voxels)	0.94	2.80	1.90	2.80	1.90
Tracing time (s)	40.63	2250.73	408.91	2158.69	89.18

Supplementary Figure 8

Combination scheme 1: *UltraTracer* combines different base tracers to achieve better performance on a 3-D confocal image stack of a Lucifer Yellow labeled human pyramidal neuron.

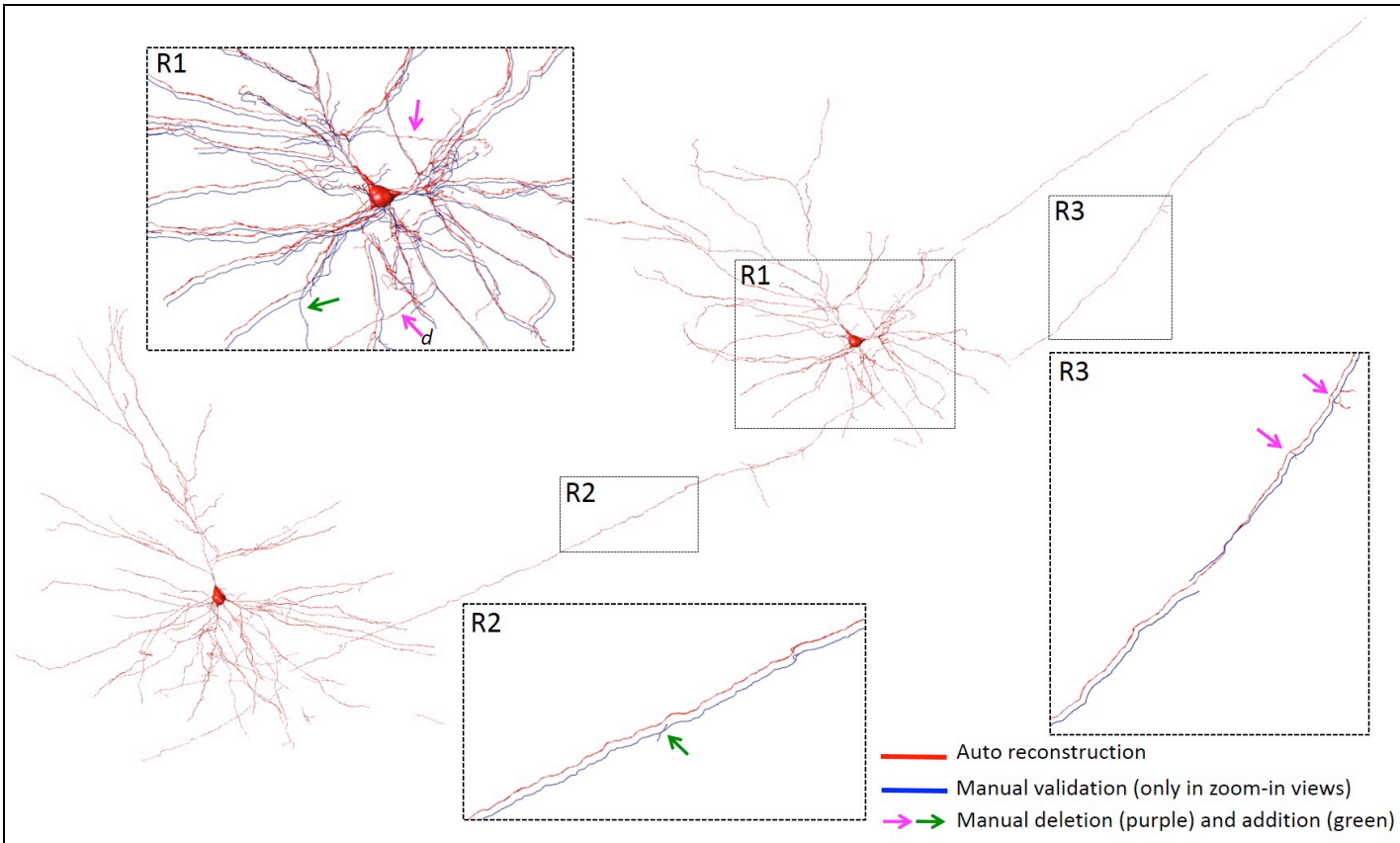
APP2+NeuTube: the soma region is traced by APP2, and the rest is traced by NeuTube. APP2+MOST: the soma region is traced by APP2, and the rest is traced by MOST. APP2+NeuTube explores 2.80 billion voxels areas, but needs 2158.69s for tracing. APP2+MOST generates a relatively complete reconstruction (1.90 billion voxels scanned areas) with a much faster tracing speed (89.18s tracing time). Neuron data used here is the neuron 4 in Supplementary Figure 3.

Algorithm	APP2	Neutube	Real-time selection	
			Best candidate	Consensus
Morphology Reconstruction				
BASDM (voxels) (against 2 ind. Manual tracings)	2.78, 3.57	5.17, 5.42	2.62, 3.42	3.73, 4.18
Tracing time (s)	115.67	3317.86	3565.69	3204.97

Supplementary Figure 9

Combination scheme 2: *UltraTracer* real-time selects suitable tracing algorithm on a confocal image stack of human pyramidal neuron.

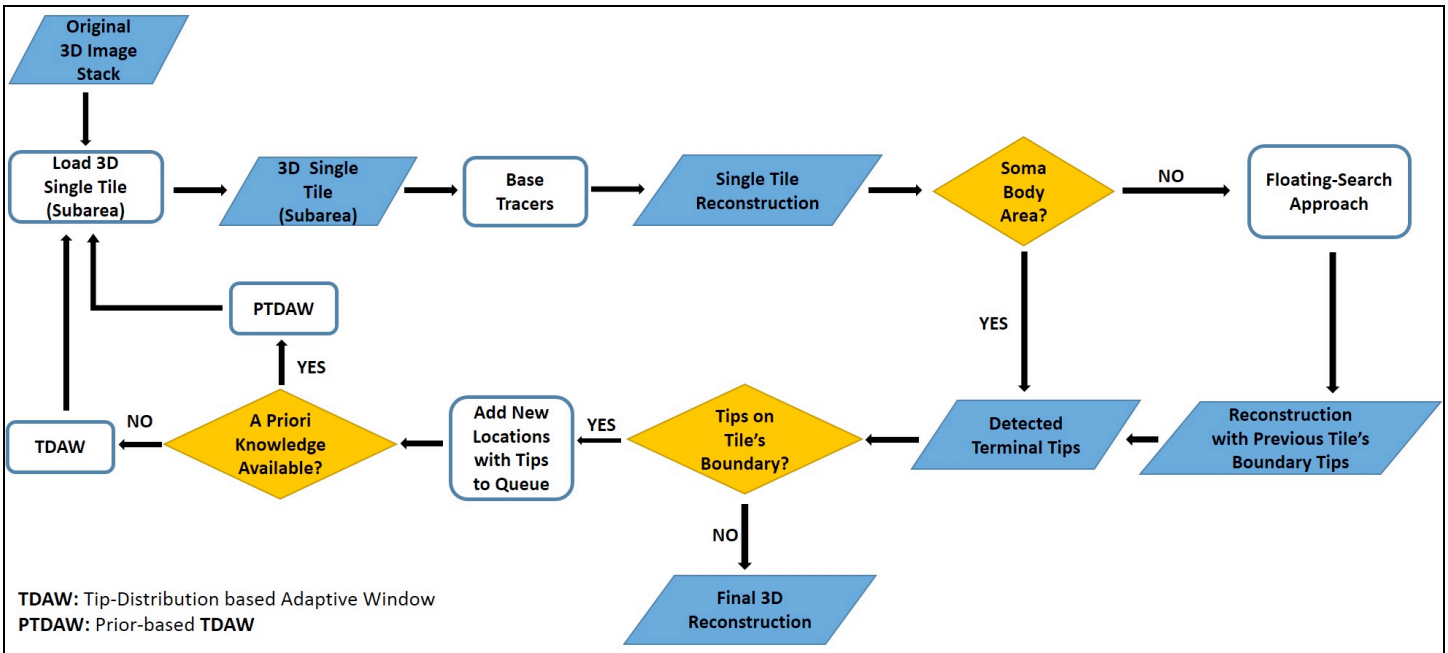
For each explored image region, two reconstructions (APP2 and NeuTube) were generated first. For the “best candidate” result, the contrast-to-background ratio in the image region around the reconstruction was used to choose the suitable algorithm. For the “consensus” result, the union of two reconstructions is used as the result for the current image region. Both two real-time selection results had similar BASDM scores (2.62 voxels and 3.42 voxels in the best candidate result, and 3.73 voxels and 4.18 voxels in the consensus result) to APP2 (2.78 voxels and 3.57 voxels) and NeuTube (5.17 voxels and 5.42 voxels). Neuron data used is the same neuron in Supplementary Figure 1B.



Supplementary Figure 10

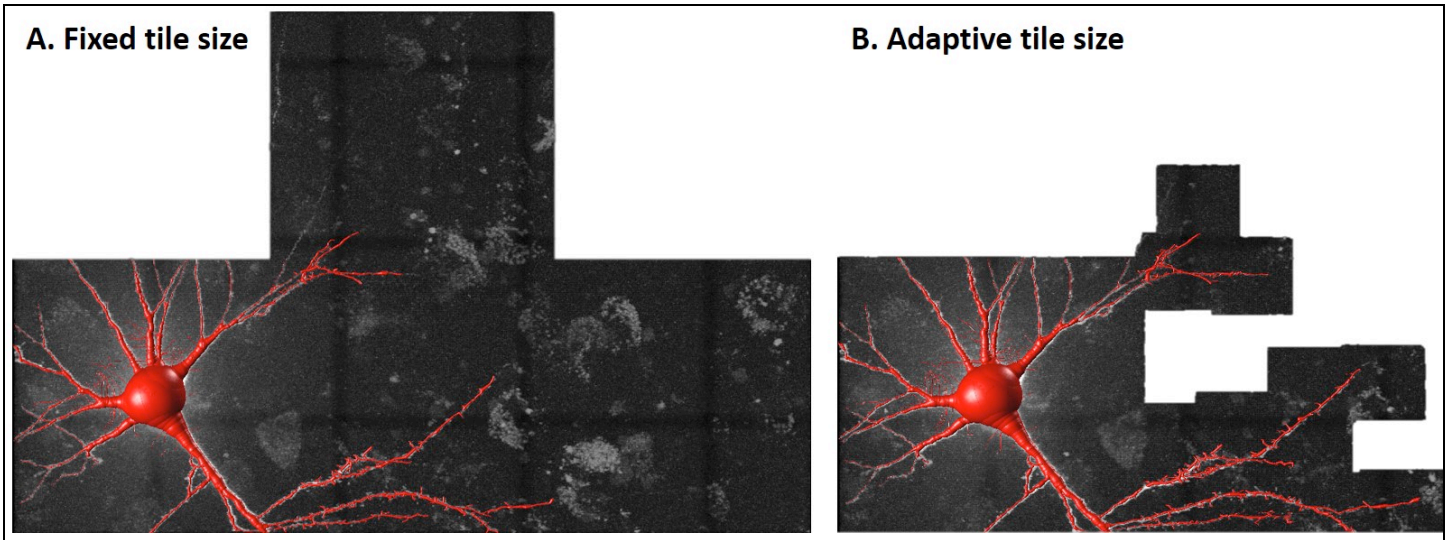
An application example of *UltraTracer* for tracing multiple biocytin-filled human neurons with axons.

The images were from 3 sections, each of which was imaged separately (voxel size 0.114μm×0.114μm× 0.28μm). *UltraTracer* was used to reconstruct automatically based on multiple starting locations on these separate image stacks. The final reconstruction (red) was assembled using the NeuronAssembler tool in Vaa3D (vaa3d.org). The reconstruction, including axons and dendrites, was also manually validated (blue in zoom-in views, slightly offset in x-direction for better visibility), with some substructures of the reconstruction edited (addition or deletion of some structures based on visual inspection). Overall more than 90% portion of the automatic reconstruction could be easily validated manually for this example, while the 10% were too difficult even for manual reconstruction (e.g. the manual deletion in location d of region R1 seemed to be a problematic deletion in the manual correction). The total lengths of the automatic and manually curated reconstructions were 22.51 and 20.15 mm, respectively.



Supplementary Figure 11

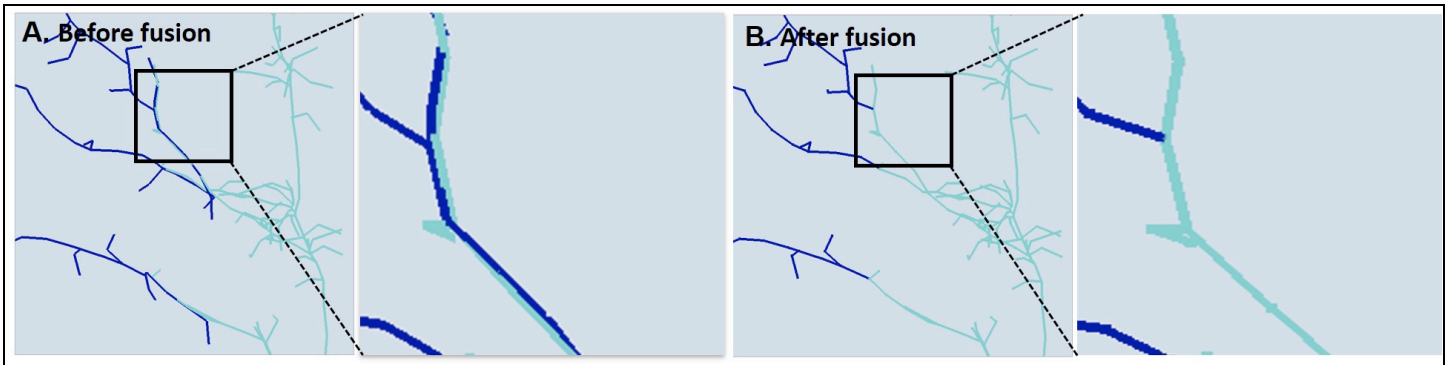
UltraTracer workflow.



Supplementary Figure 12

Fixed versus adaptive tile size.

A. Based on the boundary tips of the left tile (containing the soma), three image tiles (1.1 billion voxels) have been loaded to trace the right side of the neuron with fixed tile size. B. A much smaller part (0.55 billion voxels) of the image volume has been loaded with the adaptive tile size. The purpose of this figure is to show the comparison of loading areas to trace the right side of the neuron using fixed and adaptive tile sizes. So only the part of the traced neuron structures is shown.



Supplementary Figure 13

One reconstruction fusion example.

A. Over-tracing due to overlap between adjacent tiles. B. The over-tracing error has been fixed.

Supplementary Note:
"Automatic Tracing of Ultra-Volumes of Neuronal Images"

Hanchuan Peng [1,+,*], Zhi Zhou [1,+], Erik Meijering [2], Ting Zhao [3], Giorgio A. Ascoli [4], and Michael Hawrylycz [1]

1. Allen Institute for Brain Science, Seattle, WA, USA.
2. Erasmus University Medical Center Rotterdam, Rotterdam, Netherlands.
3. Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA.
4. Krasnow Institute for Advanced Study, George Mason University, Fairfax, VA, USA.
+ Equal contribution
* Corresponding author, Hanchuan Peng <hanchuanp@alleninstitute.org>

1. Algorithm and Key Results

1.1 Overall Algorithm

The three-dimensional (3-D) morphology of a neuron is crucial for establishing its connections and function in the context of brain circuits (Helmstaedter, 2013; Ascoli, 2015). Reconstruction of such neuron morphology from optical images is an important challenge in neuroscience (Liu, 2011, Peng, et al, 2015, Acciai, et al, 2016).

The core algorithm of *UltraTracer* (Figure 1 in main text) reconstructs a neuron structure as completely as possible from the available image data based on a formulation of maximum likelihood estimation (MLE). The underlying assumption is that the occurrence of a specific neuron structure could be modeled using the joint probability of all of its subparts given the image. Briefly, *UltraTracer* iteratively factorizes the joint probability based on progressive maximization of conditional probabilities of the occurrence of salient and continuous subparts of a neuron (**Methods**). Concretely, *UltraTracer* begins the tracing from a subarea around the soma of a neuron, typically a cube with at least 512^3 voxels. The soma could be automatically detected using a previous method or manually determined by one computer-mouse click so it was not a limiting step (**Methods**). A base tracing method T is chosen from a pool of candidate tracing methods Ω. *UltraTracer* analyzes the reconstruction produced by T and detects the tips of the neuron (Figure 1B). All such tips are added to a tip-queue. Then all tips in the tip-queue are sorted by saliency in terms of their thickness, image-intensity, and continuity (**Methods**). Next, depending on the base tracing method T, *UltraTracer* automatically and adaptively defines a new subarea to trace (Figure 1B in main text, referred to as the "tip-distribution based adaptive window" method, or TDAW), based on either the most prominent single tip, or a group of nearby tips on a polygonal face of the polyhedron of the already-traced image volume. The new reconstruction is merged onto the existing reconstruction. The already searched tips are then eliminated from the tip-queue, while new tips of the merged neuron reconstruction are added into the tip-queue. Subsequently, the tip-queue is sorted again based on saliency. The tracing procedure repeats until no new tips could be detected and the tip-queue is empty (Figure 1B in

Nature Methods doi:10.1038/nmeth.4233

main text). This way, *UltraTracer* is capable of exploring an image by following where the neurite signal goes. The final neuron morphology is produced together with the radius estimation along the reconstruction (Figure 1C in main text). In our implementation, we designed the software to quickly extract an arbitrary subvolume of interest from very large neuron image files (Methods). Therefore, *UltraTracer* can smoothly trace an image archive without the need to load a large amount of image voxels into computer memory.

1.2 Base Tracers

As a comprehensive test, in Ω we included ten representative base tracing algorithms (Supplementary Figure 1; Supplementary Figure 2; **Methods**), namely APP1 (Peng, et al, 2011), APP2 (Xiao, et al, 2013), NeuTube (Zhao, et al, 2011), MOST (Wu, et al, 2014), as well as six others including Snake (Narayanaswamy, et al, 2011), Minimum Spanning Tree (Dijkstra, 1959), NeuroGPSTree (Quan, et al, 2016), Rivulet (Liu, et al, 2016), TReMAP (Zhou, et al, 2016), and nctuTW(Lee, et al, 2012) in Supplementary Figure 2, available on the BigNeuron platform (Peng, et al, 2015; https://github.com/BigNeuron/BigNeuron-Wiki/wiki/Neuron-Reconstruction-Algorithms). These methods have different design principles and performances. The four in Supplementary Figure 1, for instance, were relatively robust, fast, and accurate (Peng, et al, 2011; Xiao, et al, 2013; Zhao, et al, 2011; Wu, et al, 2014). Despite the differences between the outputs of these methods in tracing one single or multiple tree-shape neuronal arborization patterns from one single image tile, they can all be contained in the *UltraTracer* framework (**Methods**; Supplementary Figure 1). Thus, *UltraTracer* extends different base tracing-algorithms to trace across a very large image region adaptively (Supplementary Figure 1A), a crucial utility that was not previously available to reconstruct massive scale datasets.

1.3 Efficiency, Accuracy and Scalability of *UltraTracer*

Even for an image volume with about two billion voxels, which could still be handled by some base tracers directly, *UltraTracer* reduced dramatically the total amount of required computer-memory. The performance gain of *UltraTracer* over the direct use of certain base tracers was within the range of 3 to 6 times (Supplementary Figure 1B). *UltraTracer* results were accurate as their average spatial distances to independent manual reconstructions were around 3 voxels, comparable to the spatial distance of the manual reconstructions themselves (3.56 voxels) (Supplementary Figure 1B). In addition, for two base tracers, NeuTube and MOST, *UltraTracer* had a gain of 10 to 30 folds in tracing accuracy (Supplementary Figure 1B). Testing another six base tracers (Supplementary Figure 2) within the UltraTracer framework basically indicated similar improvement, unless either the base tracer itself is too slow to produce a reconstruction with reasonable computational resource (e.g. nctuTW in Supplementary Figure 2), or the base tracer itself had been specifically designed to trace large images using a different mechanism (i.e. the TReMAP case in Supplementary Figure 2), although in this latter case the tracing quality was inferior for this dataset than obtained with *UltraTracer*. One may also note that in the specific case of Supplementary Figure 1B, the image was smaller enough for TR to load an entire image into computer memory and then use a shorter amount of time to trace than *UltraTracer*. This was because of less file IO operations used in TR. When a computer with smaller memory is used or the image volume increases greatly, TR will not be able to outperform *UltraTracer*, as shown in Supplementary Figure 3 below.

Nature Methods doi:10.1038/nmeth.4233

Since the best base tracer in Ω was APP2 in terms of speed and accuracy trade-off (Supplementary Figure 1), we further tested the APP2-based *UltraTracer* on a series of images, of which the volume ranged from 0.47 to 521.5 billion voxels (Supplementary Figure 3). *UltraTracer* was able to effectively trace only the sparse neuronal structures in these images, without spending time to analyze the entire data volumes (Supplementary Figure 3). The data volume reduction in tracing was between about 3 and almost 40 times. Particularly, *UltraTracer* was the only automatic neuron tracing method that was applicable to ultra-volumes such as neurons 5, 6 and 7 that had 118, 122 and 521 billion voxels, respectively. The traditional approach also failed for neuron 4 because the actual peak-memory requirement to trace this dataset (14-billion voxels) exceeded the total amount of available memory (128GB) in our testing machine. When the image volume increased, we observed a bigger data-volume reduction rate in tracing. This matches well with expectation, since the neuron arborizations to be traced are roughly 1-D structures while the image data is 3-D, and thus the fraction of relevant space to be explored generally decreases with increasing data volume. The results indicated robustness and scalability of *UltraTracer* for large neurons. Of note, the accuracy of reconstructions produced by *UltraTracer* was similar to that of the conventional approach, when such a traditional approach was still feasible in our testing (Supplementary Figure 3, Neurons 1, 2, and 3). Measured in terms of spatial distance, bifurcation points, and five other morphological and topological features (Falconer, 2004), and compared against the statistics drawn from collections of reconstructions produced using control-images (**Methods**), the reconstructions produced by *UltraTracer* were consistent with those generated using the traditional approach when applicable (Supplementary Figure 3, bottom-left).

In addition to confocal laser scanning images (Supplementary Figures 2 and 3), we also tested *UltraTracer* using 31 challenging brightfield images of mouse and human neurons that had distinct appearance from laser scanning images (Supplementary Figure 3, bottom-right insets; Supplementary Figure 4; Zhou, et al, 2015). After a number of tests we found these brightfield images were hard to trace successfully using the majority of automatic methods ported in BigNeuron. Differently, *UltraTracer* produced reconstructions that were consistent with visual inspection (Supplementary Figure 4). Both the peak memory and tracing time of *UltraTracer* scaled relatively smoothly on average with respect to the input image volume (Supplementary Figure 3, insets).

1.4 Using Prior Knowledge to Refine *UltraTracer*

In addition to TDAW mentioned above (**Methods**), we also considered using certain domain knowledge, or prior information, of neuron morphology to help refine the choice of the next tracing subarea. Our intuition was that a large window-size should be used for densely arborized image regions, and a small window-size would be sufficient for sparsely distributed neurites. Therefore, we estimated a lookup table of the average "expected" window size with respect to the distance between a neuron-compartment and its corresponding soma (Supplementary Figure 5A), based on analyzing the spatial distribution of 968,348 neuron-compartments in 259 manually curated human and mouse neurons in the Allen Cell Types database (http://celltypes.brain-map.org/) and the BigNeuron initiative (Peng, et al, 2015) (Supplementary Figure 6) (**Methods**). Next, we used this lookup table as the prior information to guide TDAW

Nature Methods doi:10.1038/nmeth.4233

(**Methods**). This new method, called the "prior-based TDAW" (PTDAW), enabled *UltraTracer* to trace human and mouse pyramidal neurons slightly more completely than TDAW (Supplementary Figure 5B). Quantitatively, for the human neuron in Supplementary Figure 1B, TDAW and PTDAW reconstructions were still close to each other (average spatial distance = 1.75 voxels). For a mouse pyramidal neuron (Supplementary Figure 7), we also observed similar performance of the two methods (average spatial distance = 2.85 voxels, comparable to the distances between each of these two reconstructions and the corresponding manual reconstruction, respectively 3.24 and 3.16 voxels).

1.5 Combining Multiple Base Tracers

Since *UltraTracer* was essentially a wrapper of any base neuron tracers, we also used it to combine multiple different base tracers (Supplementary Figures 8 and 9). For instance, we noted APP2 often traced well in the soma area while NeuTube and MOST were sometimes more suitable to trace curvilinear structures. Thus in one variation of *UltraTracer*, we started the APP2-tracing for the image region around soma, followed by using NeuTube or MOST for other image regions (Supplementary Figure 8). We found that in such a combination scheme, "APP2+NeuTube" was able to explore 47% larger image area than "APP2+MOST" and thus the former generated a more visually complete reconstruction, even though the latter was 24 times faster than the former (Supplementary Figure 8). In a more complicated case, for every adaptively searched image region, we profiled the reconstructions generated by several base tracers. Then we chose either the best reconstruction or their consensus as the result from the current image region (Supplementary Figure 9). These variations of *UltraTracer* could be slower or faster than some of the base tracers (e.g. the "consensus" combination was slower than APP2 but faster than NeuTube), and it could provide more consistent reconstructions compared to manual work (e.g. the combination reconstructions had better performance based on the annotator's inspection as well as roughly 20% ~ 50% smaller distance scores than that of the NeuTube results).

1.6 Tracing Fragmented Neurites of Very Large Neurons

Finally, instead of starting from a single soma location to trace one neuron, we also used *UltraTracer* to reconstruct human neurons, including their axons and dendrites, from separate but serial slices of brain tissue. We iteratively applied *UltraTracer* to multiple independent starting locations of the fragmented neuron structures, followed by stitching these fragments using Vaa3D (Peng et al, 2010) (Supplementary Figure 10). We manually validated one such example, which had totally 318.3 billion voxels in three separate sections. The total length of the tracing was 22.51mm. We found that about 90% of the compartments in the automatic reconstruction could be validated manually, while the other 10% were very challenging to reconstruct even for manual work.

Nature Methods doi:10.1038/nmeth.4233

2. Methods

2.1 UltraTracer Key Method and Implementation

A neuron structure S can be modeled as the joint occurrence of its parts S_i, i=1, …N. Given the image data D, the likelihood, i.e. the joint conditional probability, of occurrence of S is $L(S|D) = p(S|D) = p(S_1,S_2, \ldots, S_N|D)$. The optimal neuron tracing problem can be formulated as a maximum likelihood estimation (MLE) problem, i.e. maximizing $L(S|D)$ subject to constraints that the neuron's parts should be maximally connected, and the connections should be as continuous, smooth, and biologically plausible as possible. This joint probability may be factorized in a combinatorial number of ways, depending on the definition, orders, and groupings of the neuron's parts (substructures). Without loss of generality, $\max L(S|D) = \max p(S_1,S_2, \ldots, S_N|D) = \max p(S_2, \ldots, S_N|D,S_1)p(S_1|D) = \max p(S_{k+1}, \ldots, S_N|D,S_1,S_2, \ldots, S_k)p(S_2,\ldots,S_k|D,S_1)p(S_1|D)$. We used an intuitive approach to solve the MLE problem, by repeatedly finding the most probable substructures of S given the image. Obviously one such substructure should be the soma area of the neuron as well as immediately connected neurites. Then we iteratively detected other most probably connected substructures and grew the neuron reconstruction as completely as possible.

In our implementation (Supplement Figure 11), we first used the base tracer to reconstruct the cell body (soma) area, that is the most probable location to start the tracing. The size of the first subarea can be defined by the user, and the default size was 512 × 512 × 512 voxels. Then we designed a floating-search approach to smartly grow the neuron. Typically, the terminal tip close to the boundary indicates the continuity of the neuron structure. In order to assess where the neuron goes, all boundary tips from the previous tile's reconstruction were detected as the reference locations. For single-root tracing (e.g. APP1 (Peng, et al. 2011) and APP2 (Xiao and Peng 2013)), each boundary tip was used as the root input to generate a single neuron tree on the adjacent tile. Different neuron trees starting from the previous tile's boundary tip were continuously added to the adjacent tile. For multiple-segment tracing (e.g. NeuTube (Zhao, et al. 2011) and MOST (Wu, et al. 2014)), all neuron structures on the adjacent tile were traced first by the algorithm. Within the reconstruction, only the segments containing the previous tile's boundary tip were kept, and all other detected signals were removed. We also used 10% overlap with the adjacent tile to reduce the false negative rate. This floating-search approach not only solved the under-tracing problem from single-root tracing algorithms, but also eliminated the over-traced segments from multiple-segments tracing algorithms.

To efficiently explore the neuron structure, we used the density of boundary tips to adaptively define the next area, that contains the strongest signal to continue the search. First, all possible boundary tips in all six directions (left, right, up, down, in, and out) were located. In each direction, all detected boundary tips were classified into different groups based on the neighbors' distance. For each group in each direction, 1.2 × the maximum distance between two tips' locations was defined as the x, y, and z dimensions of the next area. A minimum dimension (128

Nature Methods doi:10.1038/nmeth.4233

× 128 × 128 voxels) was predetermined in case the defined dimension was too small. With the adaptive window size, *UltraTracer* loaded a much smaller amount of image volume to reconstruct a neuron (Supplementary Figure 12). This method was called tip-distribution based adaptive window (TDAW). We also introduced a variant called "prior-based TDAW" (PTDAW). In PTDAW, we used Sholl analysis (Sholl, 1953) to collect the statistics of the neuron-compartment density with respect to the soma locations (Supplementary Figure 6). Then we converted the density per 3-D unit-volume to the expected window sizes in one dimension (assuming the size of a new tracing subarea to be the same in x, y, and z). Finally, in PTDAW, the new search window size was set to be the greater one of the window size estimated using TDAW alone and the respective window size value in the lookup table, but no larger than the size of the current window containing the border-tips of consideration. The latter constraint was specifically designed for pyramidal neurons, but could be relaxed for other types of neurons.

To avoid over-tracing or topological errors due to the overlap between adjacent tiles, we designed a simple fusion approach by calculating the overlap region between two reconstruction compartments from adjacent tiles. If it was greater than 50%, only the compartment from the first traced tile was kept. Otherwise, both compartments were considered to be valid (Supplementary Figure 13). All our reconstructions were represented by a number of compartments with ID, type, coordinates, radius, and parent information. When the base tracer did not provide useful radius information, we used a Vaa3D "neuron radius" plugin to calculate the radii.

The soma, as well as other potential seed locations for starting the tracing, was automatically detected using a gray-weighted distance-transform method (Xiao and Peng, 2013), or manually determined by the virtual-finger powered one computer-mouse click technique (Peng, et al, 2014).

UltraTracer reconstructions could be further refined using Vaa3D and/or other tools. For instance, one or more human annotators can use Vaa3D's Virtual Finger functions (Peng, et al, 2014) to progressively refine a reconstruction. Crowd-labeling efforts (Kim, et al, 2014; Roskams, et al, 2016) can also generate multiple refined versions of a neuron reconstruction. Such multiple labels can be further merged using the consensus algorithms developed in the BigNeuron project (Peng, et al, 2015).

2.2 Computer Configuration

We used a Linux machine with 8 Intel(R) Xeon(R) CPU E5-1620 0 @ 3.60GHz, 128 GB memory, and C++ programming language to calculate the computational cost including peak memory and tracing time.

2.3 Software Availability

UltraTracer is available in Vaa3D software (vaa3d.org), and it is open source (https://github.com/Vaa3D/vaa3d_tools/tree/master/hackathon/zhi/neurontracer). As long as the image format can be supported by Vaa3D, it can be explored in *UltraTracer*. However, for very large-scale images (> 100 billion voxels), the computer may not have enough memory to load the

Nature Methods doi:10.1038/nmeth.4233

entire image. In that case, *UltraTracer* also supports several other image formats, specifically the Vaa3D-Terafly interface (Bria, et al, 2016) that includes 2D TIFF/Vaa3D raw files, single multipage 3D TIFF/Vaa3D raw file, three-leveled y-x-z hierarchy of tiles with 3D TIFF/Vaa3D raw files, and HDF5 volume.

2.4 Data Availability

The sample data is publicly available and can be downloaded from a GitHub link (https://github.com/Vaa3D/Vaa3D_Data/releases/download/v0.9/ultratracer_testing_data.zip).

3. Supplementary References

1. Acciai, L., Soda, P., & Iannello, G. (2016). Automated neuron tracing methods: an updated account. *Neuroinformatics*, DOI: 10.1007/s12021-016-9310-0.
2. Ascoli, G. A. (2015). *Trees of the brain, roots of the mind*. MIT Press.
3. Bria, A., Iannello, G., Onofri, L., & Peng, H. (2016). TeraFly: real-time three-dimensional visualization and annotation of terabytes of multidimensional volumetric images. *Nature Methods*, 13(3), 192-194.
4. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*. 1: 269–271. doi:10.1007/BF01386390.
5. Falconer, K. (2004). *Fractal geometry: mathematical foundations and applications*. John Wiley & Sons.
6. Helmstaedter, M. (2013). Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature Methods*, 10(6), 501-507.
7. Kim, J. S., Greene, M. J., Zlateski, A., Lee, K., Richardson, M., Turaga, S. C., ... Seung, H.S., & the EyeWirers. (2014). Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500), 331-336.
8. Lee, P. C., Chuang, C. C., Chiang, A. S., & Ching, Y. T. (2012). High-throughput computer method for 3d neuronal structure reconstruction from the image stack of the Drosophila brain and its applications. *PLoS Comput Biol*, 8(9), e1002658.
9. Liu, S.Q., Zhang, D., Li, S.D., Feng, D., Peng, H., & Cai, W.D. (2016). Rivulet: 3D neuron morphology tracing with iterative back-tracking. *Neuroinformatics*, 14(4), 387-401.
10. Liu, Y. (2011). The DIADEM and beyond. *Neuroinformatics*, *9*(2), 99-102.
11. Narayanaswamy, A., Wang, Y., & Roysam, B. (2011). 3-D image pre-processing algorithms for improved automated tracing of neuronal arbors. *Neuroinformatics*, 9(2-3), 219-231.
12. Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E., & Ascoli, G. A. (2015). BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images. *Neuron*, 87(2), 252-256.
13. Peng, H., Long, F., & Myers, E.W. (2011). Automatic 3D neuron tracing using all-path pruning. *Bioinformatics,* 27(13), i239-i247.
14. Peng, H., Ruan, Z., Long, F., Simpson, J. H., & Myers, E. W. (2010). V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nature Biotechnology*, 28(4), 348-353.

15. Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., ... & Long, F. (2014). Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature Communications*, 5: 4342, doi:10.1038/ncomms5342.
16. Quan, T., Zhou, H., Li, J., Li, S., Li, A., Li, Y., Lv, X, Luo, Q., Gong, H., & Zeng, S. (2016). NeuroGPS-Tree: automatic reconstruction of large-scale neuronal populations with dense neurites. *Nature Methods*, 13(1), 51-54.
17. Roskams, J., & Popović, Z. (2016). Power to the People: Addressing Big Data Challenges in Neuroscience by Creating a New Cadre of Citizen Neuroscientists. *Neuron*, 92(3), 658-664.
18. Sholl, D.A. (1953) Dendritic organization in the neurons of the visual and motor cortices of the cat. *J. Anat*. 87, 387–406.
19. Wu, J., He, Y., Yang, Z., Guo, C., Luo, Q., Zhou, W., Chen, S., Li, A., Xiong, B., Jiang, T., & Gong, H. (2014). 3D BrainCV: simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution. *NeuroImage,* 87, 199-208.
20. Xiao, H., & Peng, H. (2013). APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics,* 29(11), 1448-1454.
21. Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., Long, F., & Myers, E. (2011). Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, 9(2-3), 247-261.
22. Zhou, Z., Liu, X., Long, B., & Peng, H. (2016). TReMAP: Automatic 3D neuron reconstruction based on tracing, reverse mapping and assembling of 2D projections. *Neuroinformatics*, 14(1), 41-50.
23. Zhou, Z., Sorensen, S., Zeng, H., Hawrylycz, M., & Peng, H. (2015). Adaptive image enhancement for tracing 3D morphologies of neurons and brain vasculatures. *Neuroinformatics*, 13(2), 153-166.