FlyWire: online community for whole-brain connectomics

Sven Dorkenwald1,2,6, Claire E. McKellar16, Thomas Macrina1,2,6, Nico Kemnitz1,6, Kisuk Lee1,3,6, Ran Lu1,6, Jingpeng Wu1,6, Sergiy Popovych1,2, Eric Mitchell1,1,2, Barak Nehoran1,2, Zhen Jia1,2, J. Alexander Bae1,4, Shang Mu1, Dodam Ih1, Manuel Castro1, Oluwaseun Ogedengbe1, Akhilesh Halageri1, Kai Kuehner1, Amy R. Sterling1, Zoe Ashwood1,2, Jonathan Zung1,2, Derrick Brittain5, Forrest Collman5, Casey Schneider-Mizell5, Chris Jordan1, William Silversmith1, Christa Baker1, David Deutsch1, Lucas Encarnacion-Rivera1, Sandeep Kumar1, Austin Burke1, Doug Bland1, Jay Gager1, James Hebditch1, Selden Koolman1, Merlin Moore1, Sarah Morejohn1, Ben Silverman1, Kyle Willie1, Ryan Willie1, Szi-chieh Yu1, Mala Murthy1,6✉ and H. Sebastian Seung1,2,6✉

Due to advances in automated image acquisition and analysis, whole-brain connectomes with 100,000 or more neurons are on the horizon. Proofreading of whole-brain automated reconstructions will require many person-years of effort, due to the huge volumes of data involved. Here we present FlyWire, an online community for proofreading neural circuits in a Drosophila melanogaster brain and explain how its computational and social structures are organized to scale up to whole-brain connectomics. Browser-based three-dimensional interactive segmentation by collaborative editing of a spatially chunked supervoxel graph makes it possible to distribute proofreading to individuals located virtually anywhere in the world. Information in the edit history is programmatical accessible for a variety of uses such as estimating proofreading accuracy or building incentive systems. An open community accelerates proofreading by recruiting more participants and accelerates scientific discovery by requiring information sharing. We demonstrate how FlyWire enables circuit analysis by reconstructing and analyzing the connectome of mechanosensory neurons.

Electron microscopy (EM) is currently the only technique capable of reconstructing all connections in a nervous system. While the activity of large populations of neurons or even entire vertebrate brains1 can be observed via calcium imaging, adult connectomes have been mapped for only one species, Caenorhabditis elegans2,3; however, connectomes of more complex brains are now on the horizon. Part of a fly brain was imaged by EM and automatically reconstructed using deep learning. Errors in the reconstruction were corrected by 50 person-years of human proofreading to create a first draft of the hemibrain connectome. The entire fly brain connectome is of interest because of the role of Drosophila melanogaster as a model organism for circuit neuroscience. Flies are capable of a wide array of complex behaviors, including social communication, aggression, spatial navigation, decision-making and learning4–10. While the hemibrain connectome is useful for Drosophila circuit neuroscience, circuits that extend outside the hemibrain volume cannot be reconstructed (Extended Data Fig. 1).

Therefore, we have created FlyWire, an open online community for proofreading a connectome of a whole brain (flywire.ai). FlyWire is based on a previously released EM dataset of a full adult fly brain (FAFB)11. While FlyWire is dedicated to the fly brain, it introduces several methods that should be generally applicable to whole-brain connectomics. The first is a data structure called the ChunkedGraph, which is the basis for proofreading. Like previous systems12–14, FlyWire represents neurons as connected components in a graph of supervoxels (groups of voxels). A naive implementation of this underlying data structure would scale poorly to large datasets. The ChunkedGraph divides the graph spatially into chunks based on the supervoxels’ location in the dataset and adds a hierarchy of extra vertices and edges to cache information about connected components. We show that edit operations are over an order of magnitude faster than in systems relying on a naive implementation of the supervoxel graph. In addition, the ChunkedGraph enables real-time collaboration and stores the history of all edits.

FlyWire also has an open social structure. Membership in the community is open to everyone and community members immediately share the results of proofreading with each other. In contrast, another effort for reconstructing circuits from the FAFB dataset is structured as a ‘walled garden’ community15,16 and members are selected to avoid conflicts between laboratories working on the same circuit. Rather than restrict membership, FlyWire attempts to avoid conflicts by enforcing sharing of reconstructions with attribution.
The hemibrain was reconstructed through a closed proofreading process that mobilized paid workers and updated results are released to the public as the internal proofreading progresses. Our principle of openness was inspired by a previous project to reconstruct larval Drosophila circuits (A. Cardona, personal communication).

The walled garden community has historically used manual skeletonization to reconstruct neural circuits from FAFB (Extended Data Fig. 5a). As manual skeletonization is laborious, the walled garden community is starting to migrate to semi-automated reconstruction based on combining automatically generated skeletons (Extended Data Fig. 5a). FlyWire, in addition to being open, enables true three-dimensional (3D) interactive proofreading of a volumetric segmentation.

Finally, in FlyWire, the accuracy of the automated reconstruction was boosted by realigning the serial section images using deep learning (Fig. 2e,f). In the published FAFB dataset, aligned with conventional computer vision algorithms (Extended Data Fig. 5a), misalignments were numerous enough to be the dominant failure mode for automated reconstruction.

We estimate that FlyWire proofreading requires roughly 19 min of human effort per neuron. Using FlyWire we produced a complete connectivity diagram between known early mechanosensory neurons and discovered previously unknown connection patterns. FlyWire was also recently used to map the connectivity of Drosophila larvae to a persistent internal state, neurons related to olfaction and higher-order auditory neurons (Extended Data Fig. 3,080,494). Such fast response times are crucial for a globally accessible community. The above times are server response times measured during FlyWire's beta-phase (graph with n = 12,096).

Neuron segmentation. We realigned the serial section images of the FAFB dataset and generated an automated segmentation (Extended Data Fig. 2). The automatically generated segments often show many or all of the expected parts of a fly neuron: a soma, dendrites, axon terminals and a primary neurite (the usually unbranched proximal neurite connecting the soma to branching arbor downstream).

We examined reconstructions of well-known cell types before and after proofreading (Fig. 1). The automated segmentation is often accurate to begin with (quantification below) and unique morphological features across the examined cells are visible without proofreading. Qualitative comparison between images of light microscopy-level stains of the giant fiber neurons (Fig. 1, a,c,e) and a mushroom body APL neuron (Fig. 1b,d,f) show that our semi-automated segmentation procedures are able to capture large enough portions of neurons to be easily recognizable.

Chunked supervoxel graph as data structure for proofreading. Proofreading consists of two basic operations: merging falsely disconnected segments and splitting falsely merged ones. For efficient editing of automatically generated segments, we represent the segmentation as a supervoxel graph. Each graph node is a supervoxel, an atomic group of voxels that is never split (Fig. 2a,b). At any moment in time, the current segmentation is represented by the connected components of the supervoxel graph (Fig. 2c). Two segments can be merged into one by adding an edge to the graph (Fig. 2d). One segment can be split into two by removing edges (Fig. 2e,f). Users can place points on both sides of a proposed split (Fig. 2g) and our system identifies the edges that need to be removed to separate them. Our system deploys a max-flow min-cut algorithm operating on a local cutout of the supervoxel graph using predicted edge weights as capacities (Fig. 2h).

Scaling proofreading to a community demands that all users can access the latest state of the segmentation and that multiple users can work on the same neuron without introducing inconsistencies. Therefore, edits must be resolved quickly and visuals must be updated for the user. At the same time, older states of the segmentation must be accessible for review and publications. However, reads, writes and computations on the supervoxel graph can be time-consuming because they scale at least linearly with size of the components. That is because edits have global effects on the connected components even though they only introduce local changes (Fig. 2f). Because of these challenges, no system for community-based proofreading of entire neurons exists that scales to datasets as large as FAFB. Existing systems on smaller datasets restrict what proofreaders can work on or do not allow open proofreading by a community (Extended Data Fig. 3).

We designed the ChunkedGraph data structure to address these challenges (Fig. 3a). The ChunkedGraph leverages the fact that edits only change a small region of a neuron, leaving the rest unchanged. It caches information about connected components spatially, allowing it to update components rapidly after edits and restricts the part of the graph that needs to be accessed. For this, the nodes of the supervoxel graph are divided into spatial chunks (Extended Data Fig. 3). A supervoxel spanning chunk borders is carved into multiple supervoxels, each contained within a chunk. Each chunk also stores edges between the supervoxels in that chunk. We built an octree on top for storing the connected component information (Fig. 3b). In this tree, abstract nodes in higher layers represent connected components in the spatially underlying graph (Fig. 3b–d). Because the ChunkedGraph decouples regions of the same neuron from each other, regions unaffected by an edit do not need to be read and included into calculations and changes only need to propagate up the tree hierarchy (Fig. 3c and Extended Data Fig. 4). Each segment is a tree and the ChunkedGraph is a forest of all the segments.

The ChunkedGraph is initialized by ingesting the initial supervoxel graph created by our automated segmentation pipeline (Extended Data Fig. 5a). Our pipeline creates supervoxels by grouping voxels that belong to the same cell with high confidence, according to the affinity-predicting neural network (Supplementary Fig. 1) (Extended Data Fig. 5a). Edges are added to the ChunkedGraph for every pair of neighboring supervoxels in the same segment. Edge weights are also available from the automated segmentation pipeline and are ingested into the ChunkedGraph. Proofreading starts from this initial condition and proceeds by adding and subtracting edges from the ChunkedGraph.

Visualization of segments in 2D and 3D. FlyWire provides several visualizations for users to find and correct segmentation errors (Extended Data Fig. 5a). Three orthogonal two-dimensional (2D) cross-sections of the grayscale EM image are available (xy, xz and yz). 2D cross-sections of the segmentation are displayed in color and can be overlaid on the EM images. FlyWire also displays a 3D rendering (mesh) of selected segments. All of these visualizations utilize Google's Neuroglancer software (Extended Data Fig. 5a), which enables viewing of volumetric images in a web browser.

When a user interactively selects a supervoxel with a mouse click, the system rapidly displays all supervoxels belonging to the same segment within the field of view by searching the ChunkedGraph as follows. The search first traverses the tree from the selected supervoxel to the root node at the top level of the hierarchy. For mapping supervoxel to root, the server responded with a median time of 47 ms and 95th percentile of 111 ms (Fig. 3e; n = 12,096). Once the search has reached the root, it proceeds back down the tree to identify all supervoxels connected to it within the displayed area, making use of the octree structure of the ChunkedGraph. For mapping root to supervoxels, the server responded with a median time of 48 ms and 95th percentile of 465 ms per displayed chunk (n = 3,080,494). Such fast response times are crucial for a globally distributed system if every user is to see the latest state of the segmentation and no data are stored locally. The above times are server response times measured during FlyWire’s beta-phase (graph with 2.38 billion supervoxels).

Proofreading by editing the supervoxel graph. Interactive proofreading (Fig. 2g) is implemented using the ChunkedGraph as follows. The user specifies a merge by selecting two supervoxels and a mushroom body APL neuron (Fig. 1b,d,f) show that our semi-automated segmentation procedures are able to capture large enough portions of neurons to be easily recognizable.
with mouse clicks. An edge between this pair is added to the supervoxel graph (Fig. 2d and Extended Data Fig. 5). Merge edits took 940 ms at median and 1,841 ms at 95th percentile (n = 4,612) (Fig. 3f). The user specifies a split operation by selecting supervoxels with mouse clicks (Fig. 2e,g). The system applies a min-cut algorithm to remove a set of edges with minimum weight that leaves the two supervoxels in separate segments (Fig. 2g,h). Split edits had a median time of 1,818 ms and 95th percentile time of 7,137 ms (n = 2,497) (Fig. 3f).

After each edit, the ChunkedGraph generates new abstract nodes in higher layers (>1, colored nodes in Fig. 3c and Extended Data Fig. 4b). Here, the tree is only traversed in its height and not its width because connected components in neighboring regions are cached in abstract nodes. We use the same abstraction for fast-mesh generation of new components by restricting the application of costly and slow-meshing algorithms (for example marching cubes) to single chunks. We only compute meshes from the segmentation for abstract nodes on level 2 (Extended Data Fig. 3d) and then stitch

Fig. 1 | Assessing segmentation quality using known neurons. a–d, Comparison of light microscopy-level stains of giant fiber neurons (a) and a mushroom body APL neuron (b, red) to FlyWire’s AI-predicted segmentation of these cells (c,d). Arrows in c point at falsely merged pieces in the automated segmentation. e,f, The same neurons shown following proofreading. g–n, Examples of other cell types before and after proofreading (top and bottom of each image pair, respectively): central complex neurons (g,h), olfactory projection neurons (i,j), gustatory receptor neurons (k,l) and a lobula plate tangential cell (m,n). All views are frontal except for APL and central complex neurons, which are dorso-frontal. Scale bars, 30 μm (c–f,i,j); 15 μm (g,h,k,l); and 20 μm (m,n). Image for a reproduced with permission from ref. 23, Public Library of Science; image for b reproduced with permission from ref. 24, Cell Press.
these to larger components according to the hierarchy such that each abstract node up to a predefined layer has a corresponding mesh. The ChunkedGraph dynamically generates instructions for which mesh files to load for a given component.

We compared the performance of the ChunkedGraph versus an equivalent naive implementation of the supervoxel graph (Fig. 3h,i). We measured two different parts of split operations: reading of edges to compute a split and the min-cut algorithm. The ChunkedGraph benefits from being able to restrict the operations to a subregion (Fig. 3g), leading to orders-of-magnitude faster reading and calculations (Fig. 3h,i). The ChunkedGraph incurs a minor overhead only notable for very small components.

The ChunkedGraph allows concurrent and unrestricted proofreading by many users through serializing edits at a per-neuron level. Edits generate new, time-stamped nodes on higher levels (Fig. 3c and Extended Data Fig. 4b), allowing retrieval of any older state of the segmentation by applying a time filter during tree traversal. Edits can only be applied to the latest version of the segmentation. We implemented the ChunkedGraph with Google's BigTable®29, a low-latency NoSQL database. A user's ability to view a cell from any time point in the proofreading process is helpful for reviewing one's own work or the work of others (Fig. 4a). This is analogous to viewing past versions of a Wikipedia article, which are recreated using the edit history30.

Fig. 2 | Proofreading the supervoxel graph. a, Automated segmentation overlaid on EM data. Each color represents an individual putative cell. b, Different colors represent supervoxels that make up putative cells. c, Supervoxels belonging to a particular neuron, with an overlaid cartoon of its supervoxel graph. This panel corresponds to the framed square in a and the full panel in b. d, Touching supervoxels (circles) may be connected through edges in the graph indicating that they belong to the same connected component (solid lines). Merge operations add edges between supervoxels resulting in new neuronal components (orange). e, Split operations remove edges resulting in new neuronal components (blue and purple). f, Example neuron after proofreading (black). Green, blue and red components were removed during proofreading. While edit operations have global effects, edits to the supervoxel graph themselves are performed at a local level. g, For splits, users place points (red and blue dots) either 2D (left) or 3D (middle) that are linked to the underlying supervoxels (left). The proofreading back end then automatically determines which edges need to be removed and performs the split (right). The panels are screenshots from FlyWire’s Neuroglancer. The colored lines represent coordinate axes, red (x), green (y) and blue (z). h, For the operation shown in g the back end performs max-flow min-cut on the local supervoxel graph to determine the optimal cut that separates user-defined input locations (blue and purple framed circles). The thickness of the edges symbolizes the edge weight (cartoon). Scale bars, 1 μm (a–c) and 10 μm (f).
Extracting synaptic connections. With hundreds of millions of synapses in the fly brain \(^1\), automated synaptic partner identification is required for connectivity analysis at scale. Several methods have been proposed for synapse detection in large EM datasets \(^1\) but only a few solved the problem of partner assignments in polyadic synapses in the fly \(^1,15,37,38\). FlyWire should be compatible with existing and future methods that identify synaptic partners and their pre- and postsynaptic sites. Furthermore, we imported the synapses identified in a study on the whole fly brain \(^4\) into our realigned coordinate space and made them available to the community.

A fly neuron consists of a thicker, microtubule-rich ‘backbone’ and numerous thin ‘twigs’ \(^9\) (Fig. 4a). The distinction can be subjective in borderline cases, but is useful in practice. The automated segmentation contains many small ‘orphan twigs’ not assigned to any large neuronal object. Attaching orphan twigs to backbones is time-consuming and difficult because twigs contain thin processes. Therefore, we largely avoided correcting orphan twigs. The hemibrain project similarly avoids proofreading orphan twigs \(^8\). This comes at some cost; synapses involving orphan twigs will be missing from the reconstruction. Fortunately, many fly neurons are redundantly connected, with up to hundreds of synapses between a connected pair \(^5\). If omissions of synapses are statistically independent, then connections will be recalled with a probability that increases with the number of synapses involved \(^40\).

We quantified synapses missing due to orphan twigs by evaluating the segmentation at 612 randomly picked synaptic locations. For each of these synapses an expert judged whether the pre- and postsynaptic sites were at a backbone or twig (Fig. 4b–d). The haemibrain project similarly avoids proofreading orphan twigs \(^4\). This comes at some cost; synapses involving orphan twigs will be missing from the reconstruction. Fortunately, many fly neurons are redundantly connected, with up to hundreds of synapses between a connected pair \(^5\). If omissions of synapses are statistically independent, then connections will be recalled with a probability that increases with the number of synapses involved \(^40\).

We quantified synapses missing due to orphan twigs by evaluating the segmentation at 612 randomly picked synaptic locations. For each of these synapses an expert judged whether the pre- and postsynaptic reconstructions were at a backbone or twig and whether the twig was attached to a backbone or orphan (Fig. 4b–d). We found that 40.6% of all postsynaptic and 78.2% of presynaptic twigs were identified in a study on the whole fly brain \(^4\) into our realigned coordinate space and made them available to the community.
were attached to backbones. We expect our conservative proofreading to at least include all backbone and attached-twigs segments in a proofread neuron leading to an estimate of 44.6% of synapses with pre- and postsynaptic segments attached after proofreading. Hence, major connections (>9 synapses, 99.7% with at least one synapse) and most minor connections with at least 3 synapses are maintained (83% with at least 1 synapse)39,41.

For analysis, we assign synapses to neurons based on their pre- and postsynaptic coordinates (Fig. 4e) and release updated versions of the synapse table as proofreading progresses.

Quantification of proofreading effort and accuracy. To assess the effort required to proofread neuronal backbones, we proofread 183 neurons mostly with projections in early mechanosensory neuropils (antennal mechanosensory and motor center (AMMC), wedge (WED) and ventrolateral protocerebrum (VLP)). Three different people in successive rounds were instructed to proofread backbones thoroughly. The number of corrections decreased after the first round (Fig. 5a); notably large corrections (volumetric difference >1 μm³) decreased from a median of 7 in the first round to medians of 1 and 0 in the second and third round (Fig. 5b).

To quantify the impact of the different proofreading rounds further, we next compared the reconstructions before each round to their state after the third round. We calculated F1 scores with respect to volumetric completeness and correct synapse assignments (pre- and postsynaptic irrespectively) (Fig. 5c,d). One round of proofreading already recovered an accurate morphology and synapse assignment in most cells (median F1 scores: volumetric, 0.99; and synapse-based, 0.99). We then explored a faster proofreading regimen, proofreading a random subset of these cells again, focusing only on major edits. This regimen took a median of 13 min per cell while recovering accurate reconstructions (mean proofreading time, 19.1 min; median F1 scores: volumetric, 0.99; and synapse-based, 0.99; Extended Data Fig. 6).

We further assessed the quality of the automated segmentation by comparing these 183 neurons with a database of light microscopy-level images of fly neurons (FlyCircuit42) using NBLAST43 (Extended Data Fig. 7a,b). We found matches in...
FlyCircuit for 174 triple-proofread neurons. We then asked how many of FlyWire’s automated reconstructions would have sufficed to find a correct match (Extended Data Fig. 7c,d). For 70% of the unproofread segments (122 out of 174), the best hit in FlyCircuit was from the same broad cell type as the best hit after proofreading (Extended Data Fig. 7e,f). Furthermore, the exact hit was found within the top ten matches for 71% of the neurons (123 out of 174).

Researchers can proofread to their desired level of accuracy; some have reported scientific benefits without any proofreading at all. For others it may be sufficient to proofread backbones but not twigs.

Connections and subtypes in mechanosensory pathways. To validate FlyWire as a circuit discovery platform, we proofread and analyzed 178 mechanosensory neurons (belonging to seven cell classes) in the AMMC, WED and VLP neuropils in both hemispheres (Fig. 6a,b). These neurons were found based on their previously identified morphology and cell body location (Supplementary Table 1).

Airborne mechanosensory stimuli activate receptor neurons in the Johnston’s Organ (JO) of the antenna and JO neuron subtypes send broadly tonotopic projections to different zones within the AMMC48,49. AMMC neurons in turn send projections to the WED and VLP45. We identified neurons with dendrites in AMMC zones A (AMMC-A1, AMMC-A2 and giant fiber neuron (GFN)) and B (AMMC-B1 and AMMC-B2), which receive inputs largely from JO-A and JO-Bs, respectively45. Although previous work identified only 10 AMMC-B1 neurons per hemisphere45,48, we identified 59 and 58 neurons in the left and right hemisphere, respectively, all with a B1 morphology (Extended Data Fig. 8). We additionally identified neurons belonging to cell types WED-VLP (also known as iVLP-VLP45) and WV-WV (also known as iVLP-iVLP45 or WED-WED45).

AMMC-B1 neurons respond strongly to sound frequencies present in conspecific courtship songs and are thought to target WED-VLPs, based on the proximity of their processes, forming a putative pathway for courtship song processing. GFNs and AMMC-A1 neurons on the other hand, while responsive to song stimuli, are core components of the *Drosophila* escape pathway. We assessed whether any overlap between these two pathways and also looked for subtypes, based on connectivity and morphology, within each neuron class. To do this, we created a wiring diagram between all 178 identified neurons across both hemispheres (Fig. 6b,c and Extended Data Fig. 9).

Our analysis confirms previously proposed pathways between AMMC-A1 and GFN45 as well as AMMC-B1 and WED-VLPs. However, we found that only a minority of the AMMC-B1 neurons were presynaptic to WED-VLPs (left, 14 out of 59; right, 14 out of 58; Supplementary Table 1 and Fig. 6d,e); two subgroups of AMMC-B1s targeted two subgroups of WED-VLPs. This partition of WED-VLPs was directly related to input from ipsilateral AMMC-A2s (Fig. 6d) and a morphological separation of their arbors (Fig. 6f). WED-VLP-1 neurons receive convergent input from AMMC-B1-1 and ipsilateral AMMC-A2 neurons, positioning them to encode both sound stimulus motion energy (via A2) and directional sound frequency information (via B1). AMMC-B1 neurons all receive inputs from JO-B neurons, but we find they can be divided into at least five subtypes based on connectivity with other neurons (Extended Data Fig. 8 and Supplementary Table 2). AMMC-B1-1 and AMMC-B1-2 neurons target WED-VLP neurons, AMMC-B1-4 neurons target the WV-WV neurons and AMMC-B1-3 neurons send outputs to the GFN and AMMC-A1 neurons (Fig. 6e,g), suggesting the existence of cross-talk between the JO-B pathway (thought to be exclusive for processing courtship song) and the escape pathway (Fig. 6h), AMMC-B1-u (u for unidentified) neurons synapsed almost exclusively on neurons not included in our set of 178 neurons. We found that the axonal arbors of AMMC-B1-1, AMMC-B1-2 and AMMC-B1-3 striate the WED in both hemispheres, revealing how these subtypes make distinct connections (Fig. 6f). AMMC-B2 neurons receive input from ipsilateral JO-B neurons, are GABAergic and proposed to sharpen the tuning of AMMC-B1 for sound frequencies; we found that they only target AMMC-B1 neurons in the contralateral hemisphere (Fig. 6c and Extended Data Fig. 1a), suggesting a role in the spatial localization of sounds, a challenging problem for flies with their closely spaced antennal auditory receivers.

WV-WVs are GABAergic with cell bodies in the center of the brain and symmetrical processes in both hemispheres; these neurons are therefore well positioned to provide feedback inhibition within the circuit. We identified a subgroup that targets GFN, AMMC-A1 and AMMC-A2 neurons in both hemispheres (WV-WV-3) as well as a subgroup that strongly synapses onto WED-VLPs (WV-WV-1). Last, we identified a group (WV-WV-2) receiving input predominantly from AMMC-B1-2 and AMMC-B1-3 neurons but not from AMMC-B1-1 neurons (Fig. 6i). These three types of WV-WV neurons showed a correlation between the location of cell bodies and arborizations.
Fig. 6 | Connectivity between mechanosensory neurons extracted with FlyWire. a, Analysis of 178 neurons innervating three mechanosensory areas in both hemispheres; the AMMC (green) receives direct unilateral input from mechanoreceptor neurons in the JO of the antenna. b, Neurons colored by their cell type (see x and y axes of c for color mappings of individual cell types). c, Connectivity diagram between all 178 neurons ordered by cell type. Gray through lines divide cells from different hemispheres (left hemisphere (left/top); right hemisphere (right/bottom)) and colored bars separate putative cell types within each cell class. d, WED-VLP type 1 and 2 neurons, separated on the basis of differential inputs from ipsilateral AMMC-A2 neurons. e, AMMC-B1 neurons, grouped according to their outputs on to other cell types and their connectivity matrix. f, Axonal arbors of AMMC-B1 and WED-VLP subtypes in both hemispheres (insets). Arrows point to differences in arborization. g, A single AMMC-B1-4 neuron targeting a single AMMC-A1 neuron (red, AMMC-A1; turquoise, AMMC-B1-4). We found 66 automatically detected synapses from this AMMC-B1-4 neuron onto this AMMC-A1 neuron (black balls). An example synapse is shown in the EM (inset) with the arrow pointing at the T-bar. h, Connectivity diagram for mechanosensory neurons. Cell types are placed in their primary input region. i, Unpaired medial neuron types with bilateral innervation called WV-WV, separated by their connectivity with AMMC-B1 and AMMC-A1 neurons and their connectivity matrix. Scale bars, 50 μm; 10 μm (insets in f,g) and 500 nm (EM inset in g).
This analysis highlights the value of mapping connections across both brain hemispheres. It also supports the utility of EM connectomics in finding links between (previously considered distinct) pathways, understanding how functional properties of different cell types converge via connections onto common downstream cells and identifying distinctions in morphology and connectivity within known cell types.

Community organization. Users are currently being recruited from Drosophila laboratories. Professional scientists are inherently incentivized for productivity and accuracy because their own discoveries depend on their proofreading. Later on, we plan to expand recruitment to non-scientists.

During onboarding, users study self-guided training materials ('Training Materials' on https://flywire.ai, Supplementary Note 1 and Supplementary Video 1) and practice proofreading in a 'Sandbox' dataset. Users are granted proofreading privileges in the real dataset after passing an entry test. In Wikipedia, unqualified or malicious users may introduce mistakes into articles. However, even without tests the completeness and accuracy of articles in Wikipedia tends to increase over time as users detect and correct omissions or errors in articles; Wikipedia is approximately as accurate as traditional encyclopedias. FlyWire utilizes the same basic mechanism of crowd wisdom as Wikipedia, iterative collaborative editing, while adding a safety layer through entry-level testing and subsequent incentive for productivity and accuracy because their own discoveries depend on their proofreading. Later on, we plan to expand recruitment to non-scientists.

During onboarding, users study self-guided training materials ('Training Materials' on https://flywire.ai, Supplementary Note 1 and Supplementary Video 1) and practice proofreading in a 'Sandbox' dataset. Users are granted proofreading privileges in the real dataset after passing an entry test. In Wikipedia, unqualified or malicious users may introduce mistakes into articles. However, even without tests the completeness and accuracy of articles in Wikipedia tends to increase over time as users detect and correct omissions or errors in articles; Wikipedia is approximately as accurate as traditional encyclopedias. FlyWire utilizes the same basic mechanism of crowd wisdom as Wikipedia, iterative collaborative editing, while adding a safety layer through entry-level testing and subsequent incentive for productivity and accuracy because their own discoveries depend on their proofreading. Later on, we plan to expand recruitment to non-scientists.

Members must consent to follow the FlyWire community principles (https://flywire.ai), designed in consultation with the found-ers of other fly EM efforts in both larvae and adults. These efforts (including FlyWire) all require that contributors must be contacted and credited and provide an interface to retrieve contributor information. FlyWire's most important principle is openness, allowing anyone to join and (following training) edit any neuron. When using FlyWire reconstructions in a scientific publication, users must make their neurons 'public' and available to all, for which we provide a public neuron viewer (as for the neurons in this publication; Supplementary Table 2). Careful credit assignment procedures attempt to make FlyWire fair while maintaining its openness.

Discussion

FlyWire is an implementation of our proposal for an open community to proofread an automated reconstruction of the entire D. melanogaster brain. Most of the neurons analyzed here have bilateral axonal projections, but a few have unilateral projections, supporting the value of analyzing the connectome across the two hemispheres. FlyWire's completeness of the brain allows researchers to identify all partners of a neuron within the brain.

As a resource, FlyWire follows in the footsteps of other connectomics resources for D. melanogaster such as the hemibrain and the walled garden community. FlyWire builds on the openness principle of the previous project to reconstruct larval Drosophila circuits and advances over existing resources by combining this social structure with methods to enable proofreading of neurons across the whole brain.

It is likely that each whole-brain connectome will require proofreading by many people for years, in spite of increases in the accuracy of automated reconstruction. We propose that whole-brain connectomics for each animal species could benefit from a decentralized approach that crowdsources proofreading to the researchers of that species. This approach would make circuits available with zero delay, accelerating research. Researchers would be able to prioritize proofreading of their own circuits of interest and researchers could choose to proofread to any accuracy level required by their own scientific questions.

At the time of writing, over 200 researchers from over 50 laboratories have been onboarded and trained for FlyWire and membership is expanding. Using the current segmentation's mean backbone proofreading time of approximately 19 min per neuron and an estimate that the Drosophila brain contains approximately 116,000 neurons, a whole-brain connectome of these backbones with their existing twigs would require 18 person-years of proofreading assuming the use of automatic synapse detection. Ongoing improvements in both the automatic segmentation and the proofreading interface will reduce the number of errors further and make it possible to find and correct the remaining ones more rapidly. Proofreading may be sped up by future automatic detection of likely errors and suggestions of corrections.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-021-01330-0.

Received: 14 October 2020; Accepted: 25 October 2021; Published online: 23 December 2021

References

1. Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M. & Keller, P. J. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nat. Methods* **10**, 413–420 (2013).
2. White, J. G., Southgate, E., Thomson, J. N. & Brenner, S. The structure of the nervous system of the nematode Caenorhabditis elegans. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **314**, 1–340 (1986).
3. Cook, S. J. et al. Whole-animal connectomes of both Caenorhabditis elegans sexes. *Nature* **571**, 63–71 (2019).
4. Scheller, L. K. et al. A connectome and analysis of the adult Drosophila central brain. *eLife* [https://doi.org/10.7554/eLife.57443] (2020).
5. Coen, P. et al. Dynamic sensory cues shape song structure in *Drosophila*. *Nature* **507**, 233–237 (2013).
6. Duistermars, B. J., Pfeiffer, B. D., Hooper, E. D. & Anderson, D. J. A brain module for scalable control of complex, multi-motor display systems. *Neuron* **100**, 1474–1490 (2018).
7. Seelig, J. D. & Jayaraman, V. Neural dynamics for landmark orientation and angular path integration. *Nature* **521**, 186–191 (2015).
8. DasGupta, S., Ferreira, C. H. & Miesenböck, G. FoxP influences the speed and accuracy of a perceptual decision in *Drosophila*. *Science* **344**, 901–904 (2014).
9. Oswald, D. et al. Activity of defined mushroom body output neurons underlies olfactory behavior in *Drosophila*. *Neuron* **86**, 417–427 (2015).
10. Zheng, Z. et al. A complete electron microscopy volume of the brain of adult Drosophila melanogaster. *Cell* **174**, 730–743 (2018).
11. Kim, J. S. et al. Space-time wiring specificity supports direction selectivity in the retina. *Nature* **509**, 331–336 (2014).
12. Haehn, D. et al. Design and evaluation of interactive proofreading tools for connectomics. *IEEE Trans. Vis. Comput. Graph.* **20**, 2466–2475 (2014).
13. Knowles-Barley, S. et al. RhonnaNet pipeline: dense automatic neural annotation. Preprint at arXiv [http://arxiv.org/abs/1611.06973] (2016).
14. Zhao, T., Olbrich, D. J., Yu, Y. & Plaza, S. M. NeuTic software for collaborative, large-scale, segmentation-based connectome reconstruction. *Front. Neural Circuits* **12**, 101 (2018).
15. Felsenberg, J. et al. Integration of parallel opposing memories underlies memory extinction. *Cell* **175**, 709–722 (2018).
16. Dolan, M. J. et al. Communication from learned to innate olfactory processing centers is required for memory retrieval in *Drosophila*. *Neuron* **100**, 651–668 (2018).
17. Zheng, Z. et al. Structured sampling of olfactory input by the fly mushroom body. Preprint at bioRxiv [https://doi.org/10.1101/2020.04.17.041767] (2020).
18. Li, P. H. et al. Automated reconstruction of a serial-section EM Drosophila brain with flood-filling networks and local realignment. Preprint at bioRxiv [https://doi.org/10.1101/655634] (2019).
19. Mitchell, E., Keselj, S., Popovych, S., Buniyatan, D. & Sebastian Seung, H. Siamese encoding and alignment by multiscale learning with self-supervision. Preprint at arXiv [http://arxiv.org/abs/1904.02643] (2019).
20. Deutsch, D. et al. The neural basis for a persistent internal state in *Drosophila* females. *eLife* [https://doi.org/10.7554/eLife.59502] (2020).
21. Schlegel, P. et al. Information flow, cell types and stereotypy in a full olfactory connectome. *eLife* [https://doi.org/10.7554/eLife.66018] (2021).
22. Baker, C. A., McKeever, C., Nern, A. & Dorkenwald, S. Neural network organization for courtship song feature detection in Drosophila. Preprint at bioRxiv https://doi.org/10.1101/2020.08.32.2148 (2020).

23. Pérez, A. P., Jezzini, S. H., Bacon, I. P. & Blagburn, J. M. Shaking B mediates synaptic coupling between auditory sensory neurons and the giant fiber of Drosophila melanogaster. PLoS ONE 11, e0152211 (2016).

24. Wu, C.-L. et al. Heterotypic gap junctions between two neurons in the Drosophila brain are critical for memory. Curr. Biol. 21, 848–854 (2011).

25. Lee, K., Zang, J., Li, P., Jain, V. & Sebastian Seung, H. Superhuman accuracy on the SNEMI3D connectomics challenge. Preprint at arXiv https://arxiv.org/abs/1706.00120 (2017).

26. Dorkenwald, S. et al. Binary and analog variation of synapses between cortical pyramidal neurons. Preprint at bioRxiv https://doi.org/10.1101/2019.12.29.890319 (2019).

27. Zlateski, A. & Sebastian Seung, H. Image segmentation by size-dependent single linkage clustering of a watershed basin graph. Preprint at arXiv http://arxiv.org/abs/1505.00824 (2015).

28. Maitin-Shepard, J. et al. google/neuroglancer. Zenodo https://doi.org/10.5281/zenodo.5573294 (2021).

29. Chang, F. et al. Bigtable: a distributed storage system for structured data. ACM Trans. Comput. Syst. 26, 1–26 (2008).

30. Priedhorsky, R. et al. Creating, destroying, and restoring value in Wikipedia. in Proceedings of the 31st International Conference on Neural Information Processing Systems (2018).

31. Heinrich, L., Funke, J., Pape, C., Nunez-Iglesias, J. & Saalfeld, S. Synaptic cleft segmentation in non-isotropic volume electron microscopy of the complete Drosophila brain. in Medical Image Computing and Computer Assisted Intervention – MICCAI 2018 317–325 (Springer International Publishing, 2018).

32. Staffler, B. et al. SynEM, automated synapse detection for connectomics. eLife https://doi.org/10.7554/eLife.26414 (2017).

33. Schneider-Mizell, C. M. et al. Quantitative neuroanatomy for connectomics in Drosophila. eLife https://doi.org/10.7554/eLife.10059 (2016).

34. Takemura, S.-Y. et al. A visual motion detection circuit suggested by Drosophila connectomics. Nature 500, 175–181 (2013).

35. Kamikouchi, A. et al. Shaking B mediates synaptic coupling between auditory sensory neurons and the giant fiber of Drosophila melanogaster. PLoS ONE 11, e0152211 (2016).

36. Yamada, D. et al. GABAAergic local interneurons shape female fruit fly response to mating songs. J. Neurosci. 38, 4329–4347 (2018).

37. Tootoonian, S., Coen, P., Kawai, R. & Murthy, M. Neural representations of courtship song in the Drosophila brain. J. Neurosci. 32, 787–798 (2012).

38. Zung, J. et al. An error detection and correction framework for connectomics. Nat. Methods https://doi.org/10.1038/nmeth.4206 (2017).

39. Yamada, D. et al. GABAAergic local interneurons shape female fruit fly response to mating songs. J. Neurosci. 38, 4329–4347 (2018).

40. Meiernitzhagen, I. A. Of what use is connectomics? A personal perspective on the Drosophila connectome. J. Exp. Biol. https://doi.org/10.1242/jeb.164954 (2018).

41. Chiappato, A.-S. et al. Three-dimensional reconstruction of brain-wide wiring networks in Drosophila at single-cell resolution. Curr. Biol. 21, 1–11 (2011).

42. Costa, M., Manton, J. D., Ostrovsky, A. D., Prohaska, S. & Jefferis, G. S. X. E. NBLAST: rapid, sensitive comparison of neuronal structure and construction of neuron family databases. Neuron 91, 293–311 (2016).

43. Yamada, D. et al. GABAergic local interneurons shape female fruit fly response to mating songs. J. Neurosci. 38, 4329–4347 (2018).

44. Kamikouchi, A. et al. The neural basis of Drosophila gravity-sensing and hearing. Nature 458, 165–171 (2009).

45. Patella, P. & Wilson, R. I. Functional maps of mechanosensory features in the Drosophila brain. Curr. Biol. 28, 1189–1203 (2018).

46. Kim, H. et al. Wiring patterns from auditory sensory neurons to the escape and song-relay pathways in fruit flies. J. Comp. Neurol. 528, 2068–2098 (2020).

47. Clemens, J. et al. Connecting neural codes with behavior in the auditory system of Drosophila. Neuron 87, 1332–1343 (2015).

48. Azevedo, A. W. & Wilson, R. I. Active mechanisms of vibration encoding and frequency filtering in central mechanosensory neurons. Neuron 96, 446–460 (2017).

49. von Reyn, C. R. et al. A spike-timing mechanism for action selection. Nat. Neurosci. 17, 962–970 (2014).

50. Allen, M. J., Godenschwege, T. A., Tanouye, M. A. & Phelan, P. Making an escape: development and function of the Drosophila giant fiber system. Semin. Cell Dev. Biol. 17, 31–41 (2006).

51. Phelan, P. et al. Molecular mechanism of rectification at identified electrical synapses in the Drosophila giant fiber system. Curr. Biol. 18, 1955–1960 (2008).

52. Morley, E. L., Steinmann, T., Casas, J. & Robert, D. Directional cues in Drosophila melanogaster audition: structure of acoustic flow and inter-antennal velocity differences. J. Exp. Biol. 215, 2405–2413 (2012).

53. Giles, J. Internet encyclopaedias go head to head. Nature 458, 900–901 (2005).

54. Mu, S. et al. 3D reconstruction of cell nuclei in a full Drosophila brain. in Proceedings of the 31st International Conference on Neural Information Processing Systems (2017).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2021
Methods

Alignment. We started with a published-aligned dataset \(^{(14)}\) (v.14). Using a previously described method \(^{(15)}\) we trained neural networks through self-supervision to predict pairwise displacement fields between neighboring sections. Here, every location stores a vector pointing to its source location. We introduced a smoothness regularization into the training to ensure continuous transformations. This prior was relaxed at image artifacts such as cracks and folds. We first trained a convolutional neural network to detect image artifacts from a manually labeled training set, then used the predicted masks to adjust the smoothness prior before training of the displacement field network. We combined the pairwise displacement fields to generate a displacement field for every section and applied the result to the data to create a newly aligned stack.

Cross-alignment registration and brain renderings. Our alignment created a vector field for transformations from FlyWire’s space (v.14.1) to the original aligned space (v.14). To transform data from v.14 into v.14.1 (for example synapses and brain renderings), we created an inverse transformation of the vector field at a resolution of 64 × 64 × 40 nm. Locations in v.14 were transferred to v.14.1 by applying the closest displacement vector from the inverse transformation.

The v.14 brain rendering was acquired from the hemibrain website at https://flyconnectome.github.io/hemibrain/reference/hemibrain.surf.html. The v.14 whole-brain neuropil rendering was acquired from the virtual fly brain website at https://fahl.catmaid.virtuallyflybrain.org/.

Segmentation ground truth. We made use of the publicly available ground truth from the CREMI challenge (https://cremi.org) to train our convolutional neural network for predicting affinities. We realigned these ground-truth blocks as they contained misalignments as well.

Segmentation. We applied our segmentation pipeline \(^{(25,27)}\) without the use of long-range affinities. Additionally, we introduced a size-dependent threshold to break big, dumbbell-shaped mergers occurring at low threshold. In the affinity graph, we ignored any edges between two large segments \(s_1, s_2\) if mean(affinities(\(s_1, s_2\))) \(< 0.5\) and min(\(s_1, s_2\)) \(> 1,000\) and max(\(s_1, s_2\)) \(> 10,000\) representing supervoxel counts.

The ChunkedGraph proofreading back end. Supervoxel graph. The ChunkedGraph was initialized by ingesting the initial supervoxel graph created by our automated segmentation pipeline \(^{(15)}\). In this graph, every touching pair of supervoxels is connected by an edge. The weight of each edge was calculated by taking the mean of all predicted affinities from the affinity-producing neural network along the pair’s contact. Supervoxels were cut apart along chunk boundaries to ensure that they are fully contained within a chunk (Extended Data Fig. 3b,c). Pairs of supervoxels created by this cutting process were connected with infinitely strong ‘cross-chunk edges’. The initial agglomeration determined which edges are ‘on’ and ‘off’; ‘cross-chunk edges’ are always on. The connected components in the graph of ‘on’ edges represent the initial segments or ‘root objects’. Supervoxels are immutable, only the status of their edges changes and new edges might be added.

Hierarchy. In the ChunkedGraph, every connected component is represented as an octree, with the supervoxels as leaves (layer 1, L.1) and the root objects on top (root layer, LR) (Fig. 2). Nodes represent connected components in the underlying supervoxel graph. L.2 and higher nodes are connected by chunk-crossing edges forming higher layer nodes. Every node represents one connected component in the spatially underlying chunk, with nodes in higher layers representing larger chunks. A root object can have multiple connected components in any intermediate layer chunk because their connectedness might only become apparent on a higher layer.Nodes in L.1 usually have parents in L.1+ but layers might be skipped if no lateral nodes exist at a given layer. Nodes in the LR and L.2 are never skipped.

Node naming scheme. Every node is represented with an unsigned 64-bit integer. Node IDs consist of several parts: (1) The first eight bits are the layer. (2) The next three parts encode the chunk coordinate \((x, y, z)\). The size of these segments varies between layers and is usually set to the maximal number of bits needed to encode all chunk coordinates. The ChunkedGraph maintains a lookup table with layer → Nbits(\(l\)). (4) There are eight bits for a counter ID. (5) The remaining bits are used for uniqueness and together with (4) build the segment IDs.

This naming scheme ensures that all nodes from one chunk are adjacent in ID space. It grants a larger space of unique segment IDs to chunks with larger spatial extent because fewer bits are needed for the chunk coordinates in higher layers. IDs are generated by atomic counters, counting up the segment ID \((5)\). There are multiple counters per chunk, each with their own subspace \((4)\), to increase performance.

Edits and locking. Before performing an edit, the trees of the root objects affected by an edit (one or two) are locked from performing other parallel edits such that edits to the same root object are applied sequentially. Edits define edges that should either be turned ‘off’ or ‘on’ or added if not yet present. After switching edge properties, new connected components are computed in each L.2 chunk affected by the edit. These changes are propagated up the hierarchy, combining or not combining the newly formed L.2 nodes with other later nodes from the former root objects. Ultimately, a merge generates a new root node and a split generates either one or two new root nodes (a split might only generate one new root node if the removed edges did not result in a change of the global connected component).

Time stamps and versioning. Each connection between a parent and a child node is assigned a time stamp. Time stamps are generated during edits and the initial ingest. Different time stamps can be used to follow a different path through the hierarchy, with older time stamps reaching root nodes representing an earlier representation of a neuron. Root nodes represent a snapshot of a neuron in time that is valid between two edits.

Multiscale cat. To help the user perform split operations, the ChunkedGraph implements a max-flow min-cut algorithm based on sources and sinks defined by the user to find the edges that should be removed.

ChunkedGraph performance analysis. During the beta-phase of FlyWire, we measured server response times for various requests by all users (Fig. 3c-f). These numbers reflect real interactions and are affected by server and database load and are therefore an underestimate of the capability of our system.

We used real split edits as the basis for the comparison of the ChunkedGraph with proofread implementation that had been performed in FlyWire before this analysis. For this comparison, we used the same BigTable table but ignored the additional ChunkedGraph hierarchy for the naive implementation.

Proofreading front end. We adapted the Neuroglancer front end to command split and merge operations to our server back end. The FlyWire interface (Extended Data Fig. 16) extends Neuroglancer with features that support community-based proofreading. A sidebar features resources to help users get started and a global leaderboard, showing top contributors by number of edits completed in the past day or week. FlyWire updates Neuroglancer’s navigation bar with icons that fit more functionality in limited screen space, including user profile, settings, return to home view, share link generator and collapsible layer controls that allow more room for proofreading. A dataset chooser lets users switch between the Sandbox and Production data. An integrated tutorial with animations and positional pop-ups guides first-time users through the basics of viewing and editing neurons.

Proofreading evaluation. To obtain the number of edits for each neuron, we excluded edits made to chop neurons apart for inspection, which were later reversed by merging those pieces back together. We also excluded edits to a segment that was removed from that neuron later in the proofreading process. To clarify this, consider the example where a neuron was initially merged to a big component containing segments from multiple other neurons. We did not count edits for removing other neuronal segments from that component toward the edit count for the neuron at hand. More specifically, we only considered merge operations where all merge locations remained in the neuron at the end of proofreading and split operations where exactly one side of the split was contained in the final neuron.

For each final neuron, there are multiple contributing initial segments from the automated reconstruction. We selected the segment from the automated reconstruction that had the largest volumetric overlap with the neuron after three rounds of proofreading as the segment we evaluated for the automated reconstructions.

We calculated the volumetric change of edits and the volumetric completeness from the segmentation by collecting the supervoxels that were added or removed and adding up the voxels within each of them. We then multiplied this number with the nominal resolution of the segmentation \((16 \times 16 \times 40)\text{ nm}\).
a neuron was considered related to AMMC-B1 if it showed the characteristic commissure and primary neurite regardless of whether the finer backbone branches matched.

To determine how many FlyWire neurons are identifiable before proofreading, we assessed whether the FlyCircuit cell matched to the automated reconstruction belonged to the same broad cell type as the FlyCircuit cell matched to the triple-proofread reconstruction. We limited this comparison to the FlyWire neurons found to have a match in FlyCircuit.

Twig and backbone synapse evaluation. We randomly collected 99 synapses from a dataset of predicted synapses1. One expert evaluated all synapses as true-positive (615), false-positive (285) or ambiguous (99) synapses. Next, this expert evaluated the reconstructions of the pre- and postsynaptic sides of the true positive synapses as either belonging to a twig that was attached to a backbone ('twig - attached'), twig that was not attached to a backbone ('twig - orphan') or 'backbone.'

Identifying all cells within a class. We aimed to find every cell of the mechanosensory types investigated here. To do so, a location was chosen in the soma tract of a cell lineage, where proximal neurites were tightly grouped into a clear bundle, often surrounded by glia. Alternatively, in some cell types without tightly clustered proximal neurites, a location was chosen in a distinctive region of the backbone where these cells showed bundling. By examining in XY, YZ and XZ, a view was chosen that displayed the bundle in cross-section, to ensure that all cells in the bundle were visible. Every neuron in that cross-section was then examined to determine the cell type. Any dendrites detected were systematically lacked a part of their arbor (for example, AMMC-B1-1 and AMMC-A2, only one cell exists per hemisphere (151298, 69205, 1686) and (152447, 61490, 3218), right (111828, 67177, 2127) and (88476, 65205, 3043). WED-WED and AMMC-AMMC (615), false-positive (285) or ambiguous (99) synapses. Next, this expert evaluated the reconstructions of the pre- and postsynaptic sides of the true positive synapses as either belonging to a twig that was attached to a backbone ('twig - attached'), twig that was not attached to a backbone ('twig - orphan') or 'backbone.'

Proofreading neurons in FlyWire. The 183 neurons were proofread by 13 proofreaders consisting of both scientists and expert tracers from the Seung and Murthy laboratories in three rounds. Errors corrected during proofreading form two distinct categories: ‘false splits’ and ‘false merges.’ The former are locations where the automatic segmentation prematurely terminates a neuronal process, which require adding pieces to the cell, and the latter are locations where the automatic segmentation includes erroneous segments that must be removed from the cell. Proofreading efforts to locate these areas focused on the larger, microtubule-rich backbones of the neurons. Smaller, microtubule-free twigs were added if discovered incidentally while proofreading backbones, but were not actively sought out as continuations. Proofreaders first identified and corrected any large-scale errors, such as multiple distinct somata merged together. The proofreader then initiated a radial proofreading pattern of the neuron, starting from the soma, proofreading one process to completion, then returning to the initial branching point to begin the next neurite. ‘Breadcrumb’ annotations, placed along a branch and especially at fork points in the arbor, enabled proofreaders to keep track of their progress, particularly in large, dense arbors.

Proofreading relied first on the 3D morphology of the neurites, then on the 2D EM image stack for closer scrutiny when a neuron seemed morphologically suspect. Structural features that might be cause for suspicion in mammalian neurons, such as extensive self-fascilitation, were much more common in this Drosophila dataset. The idea of what constitutes ‘normal’ morphology in proofreading was updated to accommodate these characteristics and abnormal morphology was most conspicuous when viewing a cell as a whole. Multiplicity, suddenly reversed ‘flow’ of branching direction, uncharacteristically dense or sparse patches in an arbor or other instances of architectural irregularity warranted closer inspection. Smaller-scale features could also raise suspicion; abruptly truncated branches, unnaturally hard angles or smooth surfaces, large parallel backbones, narrowly pinched terminals and wide, flat, porous extensions were given extra review.

Besides inspection of the 3D cell shape, features of the 2D EM image were also used during proofreading. Detection of multiple neurites showed that the neuron could not be a single cell, which was proofread until identification was possible. We expect this approach to reveal most all cells in the bundle were visible. Every neuron in that cross-section was then examined to determine the cell type. Any dendrites detected were

Synapse proofreading and thresholding. We used a dataset of automatically detected synapses for the analysis of the morphometric connectome (Fig. 6). We filtered the synapse table with a threshold on the ‘cleft_score’ of 50.

Cell-type division by connectivity. We divided cell types into subtypes according to their connectivity and then verified the subdivision morphologically (Supplementary Table 1).

For WED-VLP, neurons receiving more than ten synapses from the ipsilateral AMMC-B2 were classified as WED-VLP-1, all others as WED-VLP-2. For AMMC-B1, selected neurons with >20 synapses onto any WED-VLP. These were then labeled as AMMC-B1-1 if they made >50% of their WED-VLP synapses onto WED-VLP and AMMC-B1-2 otherwise. Out of the remaining AMMC-B1 neurons (not AMMC-B1-1 or AMMC-B1-2), those with >80 synapses onto any WV-WV neuron were labeled as AMMC-B1-3. From the remaining AMMC-B1 cells, we labeled those as AMMC-B1-4 if they made at least 20 synapses onto AMMC-A1, AMMC-A2 and GFN combined. The remaining cells were classified as AMMC-B1-u.

For WV-WV, first, we labeled all WV-WV neurons with >20 synapses onto AMMC-A1, AMMC-A2 and GFN combined as WV-WV. Out of the remaining neurons, we labeled those with >100 synapses onto WED-VLP as WV-WV-1. WV-WV-2 was made up of all remaining WV-WV neurons.

Proofreading time calculation for a full fly brain. We based our estimate of the proofreading time for an entire fly brain on the measured mean proofreading time of 19.1 min multiplied by an estimated 116,000 neurons in the fly brain14. We assumed 2,000 h of work per year per person.
Acknowledgements
We acknowledge support from National Institutes of Health (NIH) BRAIN Initiative RF1 MH117815 to H.S.S. and M.M. M.M. further received funding through an HHMI Faculty Scholar award and an NIH R35 Research Program Award. H.S.S. also received NIH funding through RF1MH123400, U01MH117072, U01MH114824. H.S.S. also acknowledges support from the Mathers Foundation, as well as assistance from Google and Amazon. These companies had no influence on the research. We are grateful for support with FAFB imagery from S. Saalfeld, E. Trautman and D. Bock. We are grateful to D. Bock and Z. Zheng for discussions about FAFB. We thank G. Jefferis, D. Bock, A. Cardona, A. Seeds, S. Hampel and R. Wilson for advice regarding the community. We thank G. Jefferis and P. Schlegel (both with Medical Research Council Laboratory of Molecular Biology and University of Cambridge) for help with the brain renderings, transformations to FlyCircuit and NBLAST comparison with FlyCircuit neurons. We thank G. McGrath for computer system administration and M. Husseini for project administration. We are grateful to J. Martin-Shepard for Neuroglancer. We are grateful to J. Buhmann and J. Funke for discussions about their synapse resource. We thank N. da Costa, A. Bodor, C. David and the Eyewire team for feedback on the proofreading system. We thank the Allen Institute for Brain Science founder, P. G. Allen, for his vision, encouragement and support. This work was also supported by the Intelligence Advanced Research Projects Activity via Department of Interior/Interior Business Center contract no. D16PC0005 to H.S.S. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Intelligence Advanced Research Projects Activity, Department of Interior/Interior Business Center or the US Government.

Author contributions
T.M. and N.K. realigned the dataset with methods developed by E.M., B.N. and T.M. and infrastructure developed by S.P., Z.J. J.A.B., S.M. wrote code for masking defects and misalignments. K.L. trained the convolutional net for boundary detection, using ground-truth data realigned by D.J. J.W. used the convolutional net to generate an affinity map that was segmented by R.L. S.D. and N.K. created the proofreading system with input from IZ. and Z.A. N.K., M.A.C., O.O., A.H., C.S.J., K.K. and A.R.S. adapted and improved Neuroglancer for proofreading and annotations. S.D., F.C., C.S.M., C.S.J. and D. Brittain built the server infrastructure to host FlyWire and manage users. W.M.S. added the images into cloud storage. C.E.M. managed the community and trained proofreaders. C.E.M., C.J. and A.R.S. designed the training tutorials. C.E.M., C.R., J.G., D.D., I.E.R., S.K., A.B., J.H., M.M., S.M., B.S., K.W., R.W. and D. Bland tested the site and proofread neurons. C.E.M. and J.G. devised neuron annotation procedures. S.C.Y. managed proofreaders and evaluated twigs and synapses. S.D. evaluated the proofreading system. S.D. and C.E.M. analyzed the data. S.D., C.E.M., H.S.S. and M.M. wrote the manuscript. H.S.S. and M.M. led the effort.

Competing interests
T.M. and H.S.S. are owners of Zetta AI LLC, which provides neural circuit reconstruction services for research laboratories. R.L. and N.K. are employees of Zetta AI LLC.

Additional information
Extended data is available for this paper at https://doi.org/10.1038/s41592-021-01330-0.
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01330-0.
Correspondence and requests for materials should be addressed to Mala Murthy or H. Sebastian Seung.

Peer review information Nature Methods thanks Ann-Shyn Chiang, Scott Emmons and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Nina Vogt was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.
Extended Data Fig. 1 | Full brain rendering and comparison with the hemibrain. (a, b) A neuropil rendering of the fly brain (white) is overlaid with a rendering of the hemibrain and proofread reconstructions of neurons from the antennal mechanosensory and motor center (AMMC). The proofread reconstructions of (a) the AMMC-A2 neuron from the right hemisphere and (b) an WV-WV neuron are added. Scale bar: 50 μm.
Extended Data Fig. 2 | Quality of EM image alignment. (a, b) Chunked pearson correlation (CPC) between two neighboring sections in the original alignment (v14) and our realigned data (v14.1). (a) Relative change of CPC between the original and our realigned data per section. (b) Histogram of the CPC improvements from (a) (dashed red line is at 0). (c, d, e) Example images used for the CPC calculation in (a) where (c) the CPC improved through a better alignment around an artifact, (d) the CPC is almost identical and (e) the CPC overall improved due to a stretch of poorly aligned sections in the original data that were resolved in v14.1.
**Extended Data Fig. 3 | Chunking the dataset.** (a) Automated segmentation overlayed on the EM data. Each different color represents an individual putative neuron. (b) The underlying supervoxel data is chunked (white dotted lines) such that each supervoxel is fully contained in one chunk. (c) A close up view of the box in (b). (d) Application of the same chunking scheme to the meshes, requiring only minimal mesh recomputations after edits. (e) Diversity of the number of supervoxels in each chunk (median: 25661). (f) The median supervoxel contains 792 voxels. Most very small supervoxels (< 200 voxels) are the result of chunking.
Extended Data Fig. 4 | Proofreading with the ChunkedGraph. (a,) In the ChunkedGraph connected component information is stored in an octree structure where each abstract node (black nodes in levels >1) represents the connected component in the spatially underlying graph (dashed lines represent chunk boundaries). Nodes on the highest layer represent entire neuronal components. (b) Edits in the ChunkedGraph (here, a merge; indicated by the red arrow and added red edge) affect the supervoxel graph to recompute the neuronal connected components. (c) The same neuron shown in Fig. 2 after proofreading with each merged component shown in a different color. Scale bar (c): 10 μm.
Extended Data Fig. 5 | The FlyWire proofreading platform. (a) The most common view in FlyWire displays four panels: a bar with links and a leaderboard of top proofreaders (left), the EM image in grayscale overlaid with segmentation in color (second panel from left), a 3D view of selected cell segments (third panel), and menus with multiple tools (right). (b) Annotation tools include points, which can be used for a variety of purposes such as marking particular cells or synapses.
Extended Data Fig. 6 | Fast proofreading in FlyWire. Analysis of 60 neurons included in the triple proofreading analysis and fast proofreading analysis. 

(a) Comparison of the F1-Scores (0-1, higher is better; with respect to proofreading results after three rounds) between different proofreading rounds according to volumetric completeness (medians: Auto: 0.777, 1: 0.992, 2: 0.999, Fast: 0.988 means: Auto: 0.729, 1: 0.975, 2: 0.992, Fast: 0.968) and (b) assigned synapses (medians: Auto: 0.799, 1: 0.992, 2: 0.999, Fast: 0.988, means: Auto: 0.746, 1: 0.958, 2: 0.986, Fast: 0.945). ‘Auto’ refers to reconstructions without proofreading. Boxes are interquartile ranges (IQR), whiskers are set at 1.5 x IQR.
Extended Data Fig. 7 | NBLAST-based analysis of segmentation accuracy. Comparison of NBLAST matches and scores of 183 neurons before and after proofreading to assess the quality of the automated segmentation. (a) NBLAST scores of all 183 triple-proofread neurons (Fig. 5) against 16129 neurons in FlyCircuit. For each neuron in FlyWire we found the best hit in FlyCircuit according to the mean of the two NBLAST scores. (b) Scores for the best matches labeled by manual labels of match vs. no match (N(match)=174 out of 183). (c) Mean scores of the FlyWire neurons with matches before and after proofreading (N=174 neurons). (d) Histogram of the change in NBLAST score before and after proofreading. (e) Rankings of each FlyCircuit neuron matched to a triple-proofread neuron in FlyWire among the 16129 neurons before proofreading and after one round of proofreading. (f) NBLAST scores of the unproofread segments grouped by whether they matched or did not match the broad cell type after proofreading.
Extended Data Fig. 8 | Renderings of AMMC-B1 subtypes. Neurons grouped by subtype and hemisphere. AMMC, WED brain regions are shown for reference. The neuropil mesh is shown to the same scale. Scale bar: 50 μm.
Extended Data Fig. 9 | Connectivity diagrams. (a) Diagram from Fig. 6b reordered by putative subtype (b) Same diagram as in Fig. 6b with different colormap threshold.
Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- n/a Confirmed
- □ The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- □ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- □ The statistical test(s) used AND whether they are one- or two-sided
  Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- □ A description of all covariates tested
- □ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- □ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- □ For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted Give P values as exact values whenever suitable.
- □ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- □ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- □ Estimates of effect sizes (e.g. Cohen’s d, Pearson’s r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection
- All relevant repository for the presented methods are available in the seung-lab github repository, especially the ChuenkedGraph (https://github.com/seung-lab/PyChuenkedGraph) and forked version of Neuroglancer (https://github.com/seung-lab/neuroglancer) are available.

Data analysis
- Data analysis in this study was mostly limited to collecting and plotting data extracted from FlyWire. We provide a github repository with notebooks to reproduce all figures in the paper at https://github.com/seung-lab/FlyWirePaper

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. Github). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:
- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The segmentation and EM data are available through flywire ai. Neurons proofread for this study (see Fig. 6) can be viewed and accessed through the links in Sup Table 2. Synapses are made available to users who joined the FlyWire community. Synapses belonging to the cells used in this study are made available through the accompanying github repository (github.com/seung-lab/FlyWirePaper). For our comparison with FlyCircuit neurons we used the dotprops of a public dataset by Costa et al. (https://zenodo.org/record/5205616).
Field-specific reporting

Please select the one that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

√ Life sciences  □ Behavioural & social sciences  □ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size

Twirg analysis: We calculated the proportional binomial confidence interval and estimated that a sample size of 1000 lead to a 95% confidence interval of at max .03, low enough to interpret this part of the segmentation.

We gathered an overabundance of measurements for timings of the proofreading system as was possible during the beta phase. Sample sizes for the proofreading round analysis were chosen based on availability as triple proofreading is time consuming and expensive.

Data exclusions

No data was excluded.

Replication

Key measurements of the system performance and proofreading rounds were replicated by multiple repeats. The biological findings (fig. 6) were not replicated in another animal as such is not available. However, we replicated our results in the other hemisphere.

Randomization

We employed randomization where possible. For the proofreading round analysis, users were assigned to cells at random and we ensured that proofreaders did not proofread the same cell in different rounds (incl. the fast proofreading regimen).

Blinding

Blinding was not applicable as no group assignments were performed.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

| n/a | Involved in the study |
|-----|-----------------------|
| ✓   | Antibodies           |
| ✓   | Eukaryotic cell lines|
| ✓   | Palaeontology and archaeology |
| ✓   | Animals and other organisms |
| ✓   | Human research participants |
| ✓   | Clinical data         |
| ✓   | Dual use research of concern |

Methods

| n/a | Involved in the study |
|-----|-----------------------|
| ✓   | ChIP-seq              |
| ✓   | Flow cytometry        |
| ✓   | MRI-based neuroimaging |