Published in final edited form as:
*Nat Methods*. 2022 January ; 19(1): 119–128. doi:10.1038/s41592-021-01330-0.

# FlyWire: Online community for whole-brain connectomics

**Sven Dorkenwald**[1,2,*], **Claire E. McKellar**[1,*], **Thomas Macrina**[1,2,*], **Nico Kemnitz**[1,*], **Kisuk Lee**[1,5,*], **Ran Lu**[1,*], **Jingpeng Wu**[1,*], **Sergiy Popovych**[1,2], **Eric Mitchell**[1], **Barak Nehoran**[1,2], **Zhen Jia**[1,2], **J. Alexander Bae**[1,3], **Shang Mu**[1], **Dodam Ih**[1], **Manuel Castro**[1], **Oluwaseun Ogedengbe**[1], **Akhilesh Halageri**[1], **Kai Kuehner**[1], **Amy R. Sterling**[1], **Zoe Ashwood**[1,2], **Jonathan Zung**[1,2], **Derrick Brittain**[4], **Forrest Collman**[4], **Casey Schneider-Mizell**[4], **Chris Jordan**[1], **William Silversmith**[1], **Christa Baker**[1], **David Deutsch**[1], **Lucas Encarnacion-Rivera**[1], **Sandeep Kumar**[1], **Austin Burke**[1], **Doug Bland**[1], **Jay Gager**[1], **James Hebditch**[1], **Selden Koolman**[1], **Merlin Moore**[1], **Sarah Morejohn**[1], **Ben Silverman**[1], **Kyle Willie**[1], **Ryan Willie**[1], **Szi-chieh Yu**[1], **Mala Murthy**[1,†], **H. Sebastian Seung**[1,2,†]

[1]Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA

[2]Computer Science Department, Princeton University, Princeton, NJ, USA

[3]Electrical Engineering Department, Princeton University, Princeton, NJ, USA

[4]Allen Institute for Brain Science, Seattle, WA, USA

[5]Brain & Cognitive Sciences Department, Massachusetts Institute of Technology, Cambridge, MA, USA

## Abstract

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 3D interactive segmentation

[†] Correspondence to sseung@princeton.edu and mmurthy@princeton.edu.
[*]These authors contributed equally

CONTRIBUTIONS
TM, NK realigned the dataset with methods developed by EM, BN, TM and infrastructure by SP, ZJ, JAB, SM wrote code for masking defects and misalignments. KL trained the convolutional net for boundary detection, using ground truth data realigned by DI. JW used the convolutional net to generate an affinity map which was segmented by RL. SD, NK created the proofreading system with input from JZ and ZA. NK, MAC, OO, AH, CSJ, KK, ARS adapted and improved Neuroglancer for proofreading and annotations. SD, FC, CSM, CSJ, DeB built the server infrastructure to host FlyWire and manage users. WMS ingested the images into cloud storage. CEM managed the community and trained proofreaders. CEM, CJ, ARS designed the training tutorials. CEM, CB, JG, DD, LER, SK, AB, JH, MM, SM, BS, KW, RW, DoB tested the site and proofread neurons. CEM, JG devised neuron annotation procedures. SCY managed proofreaders and evaluated twigs and synapses. SD evaluated the proofreading system. SD, CEM analyzed the data. SD, CEM, HSS, MM wrote the manuscript. HSS, MM led the effort.

Competing interests
TM and HSS are owners of Zetta AI LLC, which provides neural circuit reconstruction services for research labs. RL and NK are employees of Zetta AI LLC.

Disclaimer: 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 IARPA, DoI/IBC, or the U.S. Government.

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 programmatically 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 analysing the connectome of mechanosensory neurons.

## INTRODUCTION

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 brains [1] can be observed via calcium imaging, adult connectomes have been mapped for only one species, *C. elegans* [2,3]. However, connectomes of more complex brains are now on the horizon. A milestone has been the recent release of a *Drosophila* hemibrain connectome [4]. 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 would be 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 learning [5–9]. While the hemibrain connectome is useful for *Drosophila* circuit neuroscience, circuits that extend outside the hemibrain volume cannot be reconstructed (Extended Figure 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) [10]. 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 systems [11–14], 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. 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” community [15,16], and members are selected to avoid conflicts between labs 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 [4]. 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 [15,16]. Since manual skeletonization is laborious, the walled garden community is starting to migrate to semi-automated reconstruction [17] based on combining automatically generated skeletons [18]. FlyWire, in addition to being open, enables true 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 [19]. In the published FAFB dataset, aligned with conventional computer vision algorithms [10], misalignments were numerous enough to be the dominant failure mode for automated reconstruction.

We estimate that FlyWire proofreading requires roughly 19 minutes 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* neurons related to a persistent internal state [20] and higher-order auditory neurons [21].

# RESULTS

## Neuron segmentation

We realigned the serial section images of the FAFB dataset[10], 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 arbors 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 [22] (Fig. 1 a,c,e) and a mushroom body APL neuron[23] (Fig. 1 b,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 the 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 the 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 [11] or do not allow open proofreading by a community [4].

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 build 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, 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 [24–26]. Our pipeline creates supervoxels by grouping voxels that belong to the same cell with high confidence, according to the affinity-predicting neural network (Supplementary Figure 1) [26]. 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 2D cross sections of the grayscale EM image are available (xy, xz, 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 [27], 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 with mouse clicks. An edge between this pair is added to the supervoxel graph (Fig. 2d, Extended Data Fig. 5). Merge edits took 940 ms at median, and 1841 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 (e.g. marching cubes) to single chunks. We only compute meshes from the segmentation for abstract nodes on level 2 (Extended Fig. 3d) and then stitch 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 on a per-neuron level. Edits generate new, timestamped nodes on higher levels (Fig. 3c, Extended Data Fig. 4b), allowing the 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 [28], a low-latency NoSQL database. A user's ability to view a cell from any timepoint 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 history [29].

## Extracting synaptic connections

With hundreds of millions of synapses in the fly brain [30], automated synaptic partner identification is required for connectivity analysis at scale. Several methods have been proposed for synapse detection in large EM datasets [30–35] but only a few solved the problem of partner assignments in polyadic synapses in the fly [30,34,36,37]. 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[30] 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" [38] (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 [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 [39]. If omissions of synapses are statistically independent, then connections will be recalled with a probability that increases with the number of synapses involved [38].

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 attached to backbones. We expect our conservative proofreading to at least include all backbone and attached-twig 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 one synapse) [38,40].

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 > 1µm$^3$) 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, 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 minutes per cell while recovering accurate reconstructions (mean proofreading time: 19.1 minutes, median F1 Scores: volumetric: 0.99, 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 (FlyCircuit[41]) using NBLAST[42](Extended Data Fig. 7a,b). We found matches in FlyCircuit for 174 triple-proofread neurons. We asked for how many of these FlyWire's automated reconstruction 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). Further, the exact hit was found within the top 10 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[20].

## 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 [43–47] (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 AMMC [47,48]. AMMC neurons in turn send projections to the WED and VLP [44]. We identified neurons with dendrites in AMMC zones A (AMMC-A1, AMMC-A2, GFN (giant fiber neuron)) and B (AMMC-B1, AMMC-B2), which receive inputs largely from JO-As and JO-Bs respectively [49]. Although prior work identified only 10 AMMC-B1 neurons per hemisphere [44,49], 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 (aka iVLP-VLP [44]) and WV-WV (aka iVLP-iVLP [44] or WED-WED [50]).

AMMC-B1 neurons respond strongly to sound frequencies present in conspecific courtship songs [51] and are thought to target WED-VLPs, based on the proximity of their processes [44], forming a putative pathway for courtship song processing. GFNs and AMMC-A1 neurons on the other hand, while responsive to song stimuli [43,50], are core components of the *Drosophila* escape pathway [52,53]. We assessed whether there was 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, Extended Data Fig. 9).

Our analysis confirms previously proposed pathways between AMMC-A1 and GFN [54] as well as AMMC-B1 and WED-VLPs [44]. However, we found that only a minority of the AMMC-B1 neurons innervated WED-VLPs (left: 14 out of 59, right: 14 out of 58, Supplementary Table 1, 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) [51].

AMMC-B1 neurons all receive inputs from JO-B neurons [49], but we find they can be divided into at least 5 subtypes based on connectivity with other neurons (Extended Data Fig. 8, Supplementary Table 2). AMMC-B1–1 and AMMC-B1–2 neurons project to WED-VLP neurons, AMMC-B1–4 neurons target only 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, −2 and −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 [46]; we found that they only target AMMC-B1 neurons in the contralateral hemisphere (Fig. 6c, Extended Fig. 1a), suggesting a role in the spatial localization of sounds, a challenging problem for flies with their closely spaced antennal auditory receivers [55].

WV-WVs are GABAergic [44] 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). Lastly, 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.

This analysis highlights the value of mapping connections across both brain hemispheres and 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* labs. 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, 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 [56]. 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 spot checks of proofreading quality.

Members must consent to follow the FlyWire community principles (https://flywire.ai), designed in consultation with the founders of other fly EM efforts in both larva and adult. 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 *Drosophila 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 *Drosophila melanogaster* such as the hemibrain [4] 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.

Using the current segmentation's mean backbone proofreading time of approximately 19 minutes per neuron, and an estimate that the *Drosophila* brain contains approximately 100,000 cells, a whole-brain connectome of these backbones with their existing twigs would require 16 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 [57].

At this writing, over 160 researchers from over 40 labs have been onboarded and trained for FlyWire, and membership is expanding. There are hundreds of labs studying *Drosophila* neural circuits worldwide, and the *Drosophila* research community has a long history of sharing and collaboration. Furthermore, the automated segmentation is now so accurate that research questions can be answered by only modest proofreading effort.

# METHODS

## Alignment

We started with a published aligned dataset [10] (v14). Using a previously described method [19] 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 during 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 (v14.1) to the original alignment space (v14). In order to transform data from v14 into v14.1 (e.g. synapses and brain renderings), we created an inverse transformation of the vector field at a resolution of 64 × 64 × 40 nm. Locations in v14 were transferred to v14.1 by applying the closest displacement vector from the inverse transformation.

The v14 brain rendering was acquired from the hemibrain website: https://flyconnectome.github.io/hemibrainr/reference/hemibrain.surf.html. The v14 whole brain neuropil rendering was acquired from the virtual fly brain website: https://fafb.catmaid.virtualflybrain.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 [24–26] without the use of long-range affinities. Additionally, we introduced a size-dependent threshold to break big, dumbbell shaped mergers occuring at low threshold. In the affinity graph, we ignored any edges between two large segments *s1*, *s2* if $mean(affinities(s1, s2)) < 0.5$ and $min(s1, s2) > 1{,}000$ and $max(s1, s2) > 10{,}000$ representing supervoxel counts.

## The ChunkedGraph proofreading backend

**Supervoxel graph.—**The ChunkedGraph was initialized by ingesting the initial supervoxel graph created by our automated segmentation pipeline [25]. 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 they are fully contained within a chunk (Extended Data Fig. 3 b,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, L1) and the root objects on top (root layer, LR) (layer 5 in Figure 3b). L2 nodes represent connected components in the underlying supervoxel graph. L2 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 Lx usually have parents in Lx+1 but layers might be skipped if no lateral nodes exist at a given layer. Nodes in the LR and L2 are never skipped.

**Node naming scheme.—**Every node is represented with an unsigned 64-bit integer. Node IDs consist of 6 parts. (1) The first 8 bits are reserved for 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 → N(bits). (4) 8 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 L2 chunk affected by the edit. These changes are propagated up the hierarchy, combining or not combining the newly formed L2 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).

**Timestamps and versioning.—**Each connection between a parent and a child node is assigned a timestamp. Timestamps are generated during edits and the initial ingest. Different timestamps can be used to follow a different path through the hierarchy, with older timestamps 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.

**Multipoint cut.—**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. 3e,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 a naive implementation that had been performed in FlyWire’s beta phase prior to this analysis. For this comparison, we used the same BigTable table but ignored the additional ChunkedGraph hierarchy for the naive implementation.

## Proofreading frontend

We adapted the Neuroglancer Frontend to command split and merge operations to our server backend. The FlyWire interface (Extended Data Fig. 5a) 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, that 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 merged to a big component containing segments from multiple other neurons. We did not count edits for removing other neuronal segments from that component towards 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 × 16 × 40 nm).

## NBLAST-based segmentation quality analysis and comparison to FlyCircuit

We gathered skeleton locations of 16129 cells in FlyCircuit [42,58]. In this set, all cells had been mirrored to the left side of the brain if their cell body was located in the right hemisphere.

We computed skeletons for 183 triple-proofread neurons and their versions throughout proofreading (Auto, round 1, round 2, round 3) using pcg_skel (https://github.com/AllenInstitute/pcg_skel, with invalidation_d=2), which uses the ChunkedGraphs structure to generate skeletons. Next we used navis to map all skeletons to the left hemisphere in accordance with the FlyCircuit data before transforming them into the FlyCircuit space (FCWB). navis is based on natverse (https://github.com/schlegelp/navis)[59]. Using navis we computed NBLAST scores for all transformed FlyWire skeletons. We computed forward and backward scores as well as their mean. We used the mean score for ranking matches in FlyCircuit.

For each triple-proofread neuron in FlyWire, we assessed the best matching neuron in FlyCircuit according to the mean NBLAST score. In a review step, a cell from FlyWire was manually assessed to match a cell from FlyCircuit if they were recognizable as belonging to the same broad cell type (e.g., WV-WV), without making finer distinctions between subtypes (N=174 of 183). For example, 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 999 synapses from a dataset of predicted synapses [30]. 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 find the desired cell type. Any neuron that could not be classified was proofread until identification was possible. We expect this approach to reveal most or all cells within a lineage, however there could be reasons why some might be missed (such as a proximal neurite that travels outside the bundle). Locations used: AMMC-A1, right: (103406, 54035, 4640), left: (159748, 56141, 3678). AMMC-B2 commissure: (132104, 71166, 3416). WED-VLP, left: (172786, 69380, 2254), right: (88476, 65205, 3043). WED-WED and AMMC-AMMC (same midline soma tract): (132008, 84118, 4272). AMMC-B1: not all were tightly bundled, so two locations were used per hemisphere for cross-sections: left: (151298, 69205, 1686) and (152447, 61490, 3218), right: (111828, 67177, 2127) and (111214, 60441, 3615). GFN and AMMC-A2: only one cell exists per hemisphere (confirmed for AMMC-A2 by examining all other neurons within its commissure).

We noticed that some of the AMMC-B1 neurons in the left hemisphere systematically lacked a part of their arbor (e.g., see AMMC-B1–1 and −2 in Extended Data Fig. 8). This could not be attributed to errors in the segmentation or artifacts during the imaging process, and may be due to a developmental deformity in this small region of this fly’s brain. Apart from this deformity, the connectivity and morphology of these neurons appeared to be similar to the corresponding neurons in the other hemisphere.

## Synapse proofreading and thresholding

We used a dataset of automatically detected synapses [30] for the analysis of the mechanosensory connectome (Fig. 6). We filtered the synapse table with a threshold on the “cleft_score” of 50.

During analysis, we noticed a higher occurrence of false positive synapses between some cell types. These were usually cell types that had a high number of contacts due to spatial proximity. We randomly inspected about 25 synapses per <cell type> to <cell type> (e.g. AMMC-B1 to WED-VLP or AMMC-B1 to AMMC-B1) and disregarded connections with mostly false positive or questionable synapses. This exclusion mostly affected connections within cell types (e.g. AMMC-B1 to AMMC-B1). We did not remove single false positive synapses; the remaining <cell type> to <cell type> connections reported in Figure 6 still have false positive synapses among them.

## Cell type division by connectivity

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

**WED-VLP:** Neurons receiving more than 10 synapses from the ipsilateral AMMC-A2 were classified as WED-VLP-1, all others as WED-VLP-2.

**AMMC-B1:** We first selected neurons with more than 30 synapses onto any WED-VLP. These were then labeled as AMMC-B1–1 if they made more than 50% of their WED-VLP synapses onto WED-VLP-1 and AMMC-B1–2 otherwise. Out of the remaining AMMC-B1 neurons (not −1 or −2), those with more than 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 cells combined. The remaining cells were classified as AMMC-B1-u.

**WV-WV:** First, we labeled all WV-WV neurons with more than 20 synapses onto AMMC-A1, AMMC-A2 and GFN combined as WV-WV-3. Out of the remaining neurons, we labeled those with more than 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 minutes multiplied with an estimated 100,000 neurons in the fly brain. We assumed 2000h of work per year and person.

## Proofreading neurons in FlyWire

183 neurons were proofread by 13 proofreaders consisting of both scientists and expert tracers from the Seung and Murthy labs 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 which must be removed from the cell. Proofreading efforts to locate these areas focused largely 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 forking 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 an area appeared morphologically suspect. Structural features that might be cause for suspicion in mammalian neurons, such as extensive self-fasciculation, 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. Multipolarity, 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 for proofreading. Determining what constitutes a segment’s border was crucial when extending a false split, and special attention was paid to features that might disrupt or obscure the border of a segment, such as cell membranes that were parallel with the direction of the slice plane (the Z-direction). Tracking an endoplasmic reticulum (ER) tubule or microtubule was often used to confirm continuations in these cases, while a sudden change in the overall direction of microtubule flow could indicate a false merge. The apparent darkness of a neurite’s cytosol, the presence and size of any vesicle clouds, and the appearance of other intracellular organelles were also used to verify the continuation of a segment. When a cell’s proofreading was complete, its general shape was validated against other neurons of the same type.

After thorough proofreading of neurons by experts, we then evaluated whether neurons could be proofread more quickly while still producing acceptable quality (referred to as “Fast” proofreading in Extended Data Fig. 6). Experts were given neurons they had not previously worked on, in an unedited state, and instructions to look only for major edits. (For example, only fix accidental mergers if they would cut off a significant piece of backbone.) Proofreaders were instructed to skip edits that were very time-consuming to resolve, and to skip accidental mergers with small pieces of glia.

## Data availability

FlyWire's EM data and unproofread segmentation are publicly available. FlyWire's proofread segmentation is available to the community first as outlined in FlyWire's principle. Published proofread neurons are publicly available. FlyWire's website (flywire.ai) describes how to access these different data sources.

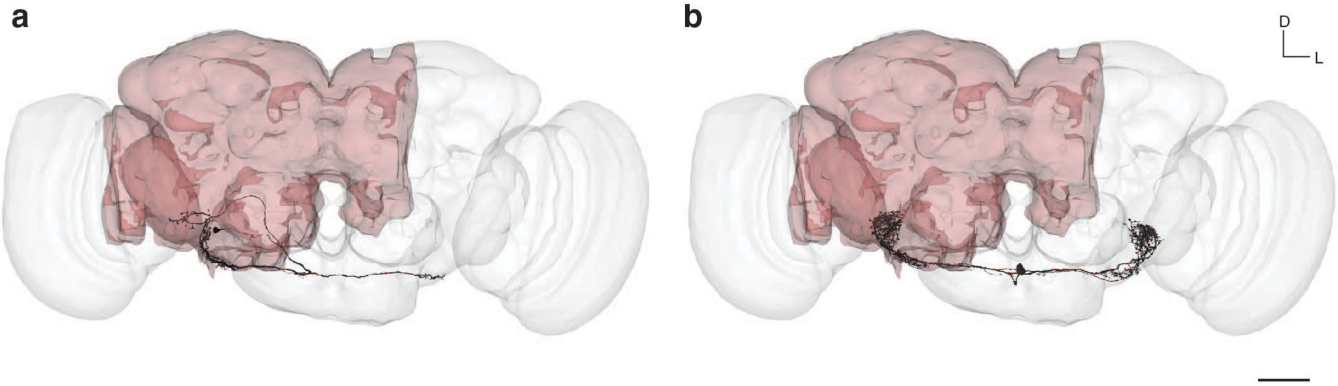
All neuron reconstructions used in this manuscript are available and linked in Supplementary Table 2. Additionally, all data necessary to reproduce the analyses in this manuscript are available through the data analysis github repository (https://github.com/seung-lab/FlyWirePaper). This includes the connectivity map between all neurons included in the mechanosensory analyses.

For the comparison with FlyCircuit neurons we used the dotprops of a public dataset [58] (https://zenodo.org/record/5205616).

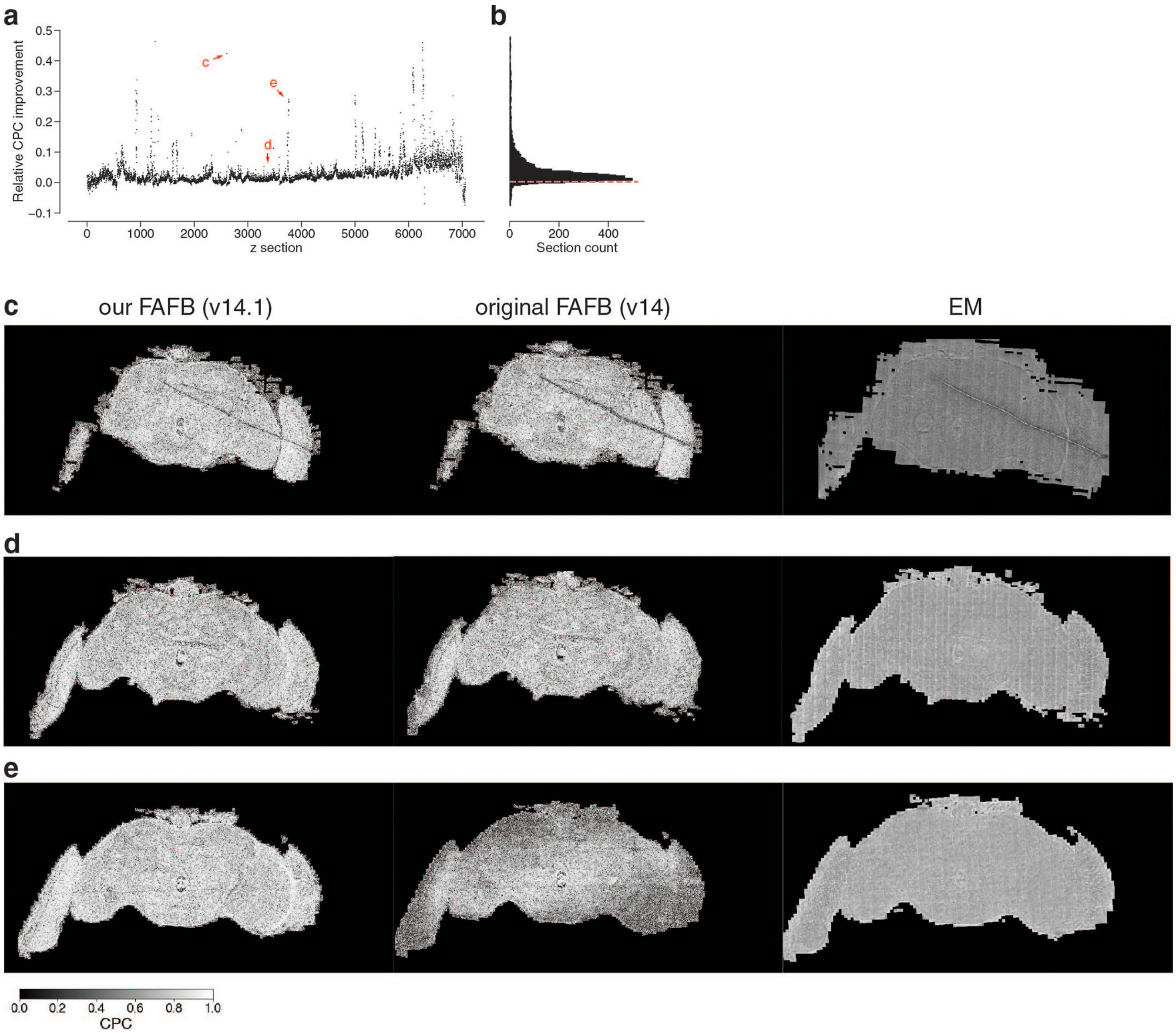
## Code availability

All repositories presented in this manuscript are open-sourced and available through the seung-lab github project. Specifically, our implementation of the ChunkedGraph is available there (https://github.com/seung-lab/PyChunkedGraph). Further, the code to reproduce all figures in this manuscript is available on github as well (https://github.com/seung-lab/FlyWirePaper).

## Extended Data

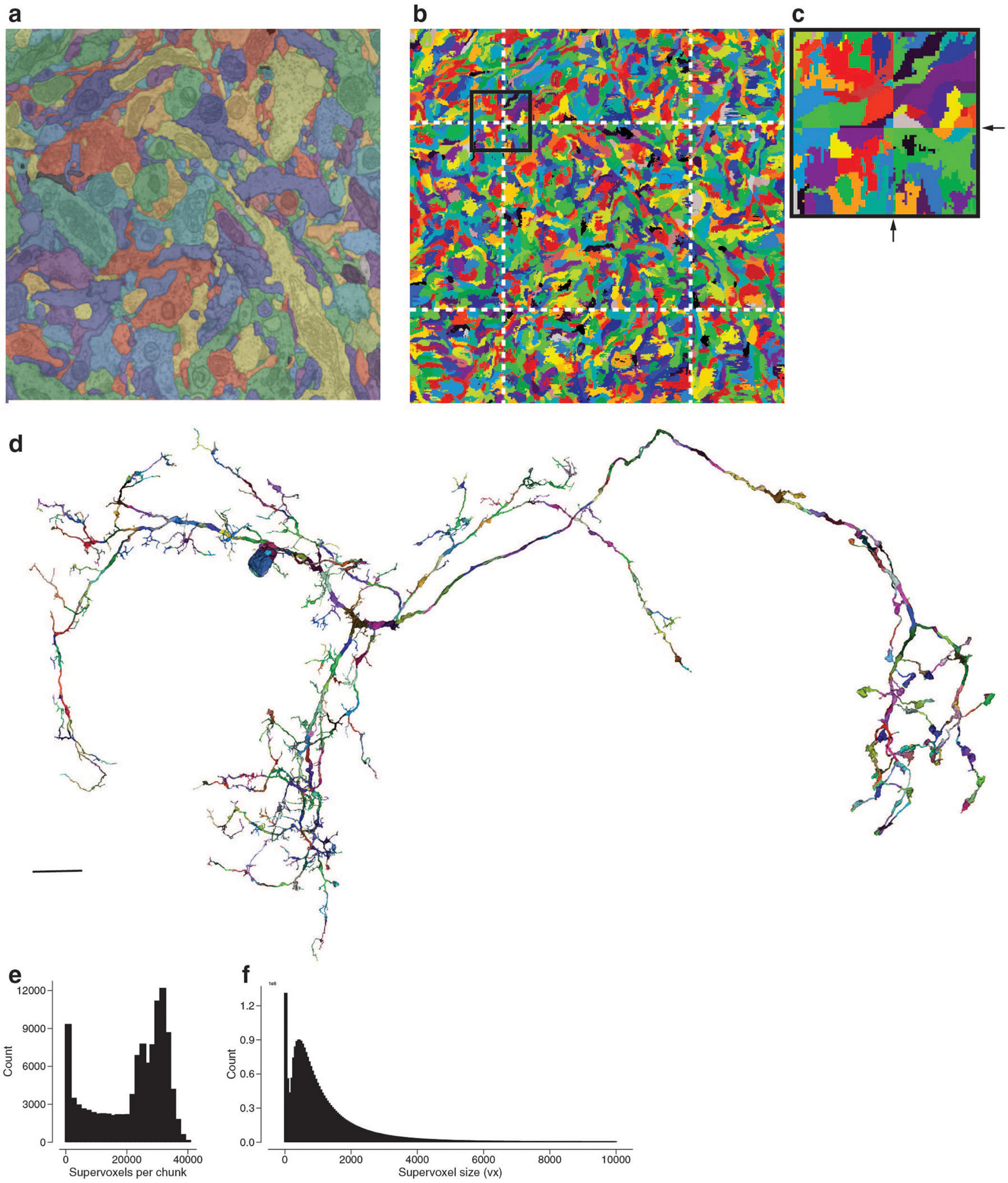


**Extended Data Fig. 1: Full brain rendering and comparison with the hemibrain.**
**(a, b)** A neuropil rendering of the fly brain (white) is overlaid with a rendering of the hemibrain and proofread reconstructions of neurons from the antennal mechanosensory and motor center (AMMC). The proofread reconstructions of **(a)** the AMMC-A2 neuron from the right hemisphere and **(b)** an WV-WV neuron are added. Scale bar: 50 μm



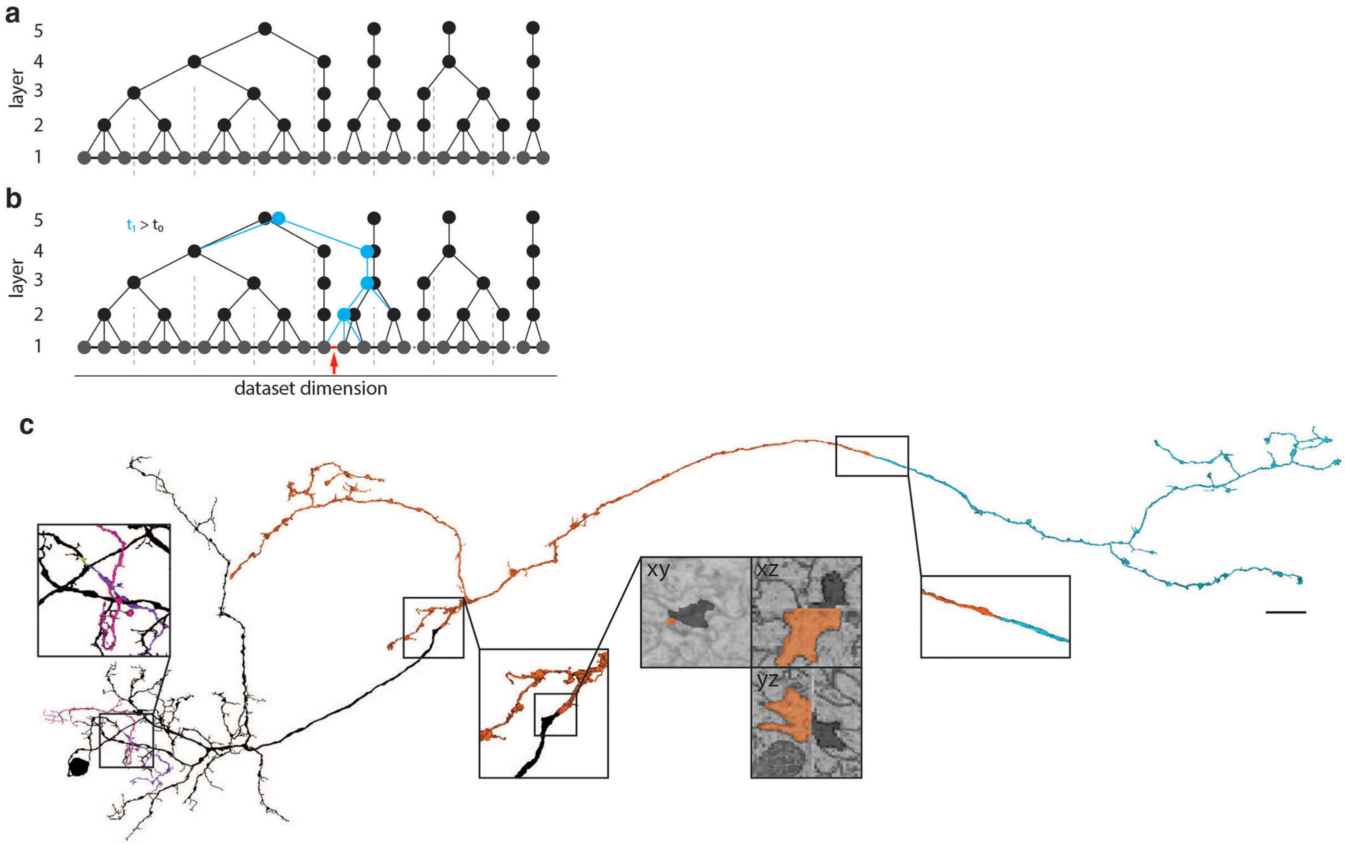
**Extended Data Fig. 2: Quality of EM image alignment.**

**(a, b)** Chunked pearson correlation (CPC) between two neighboring sections in the original alignment (v14) and our re-aligned data (v14.1). **(a)** Relative change of CPC between the original and our re-aligned 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.



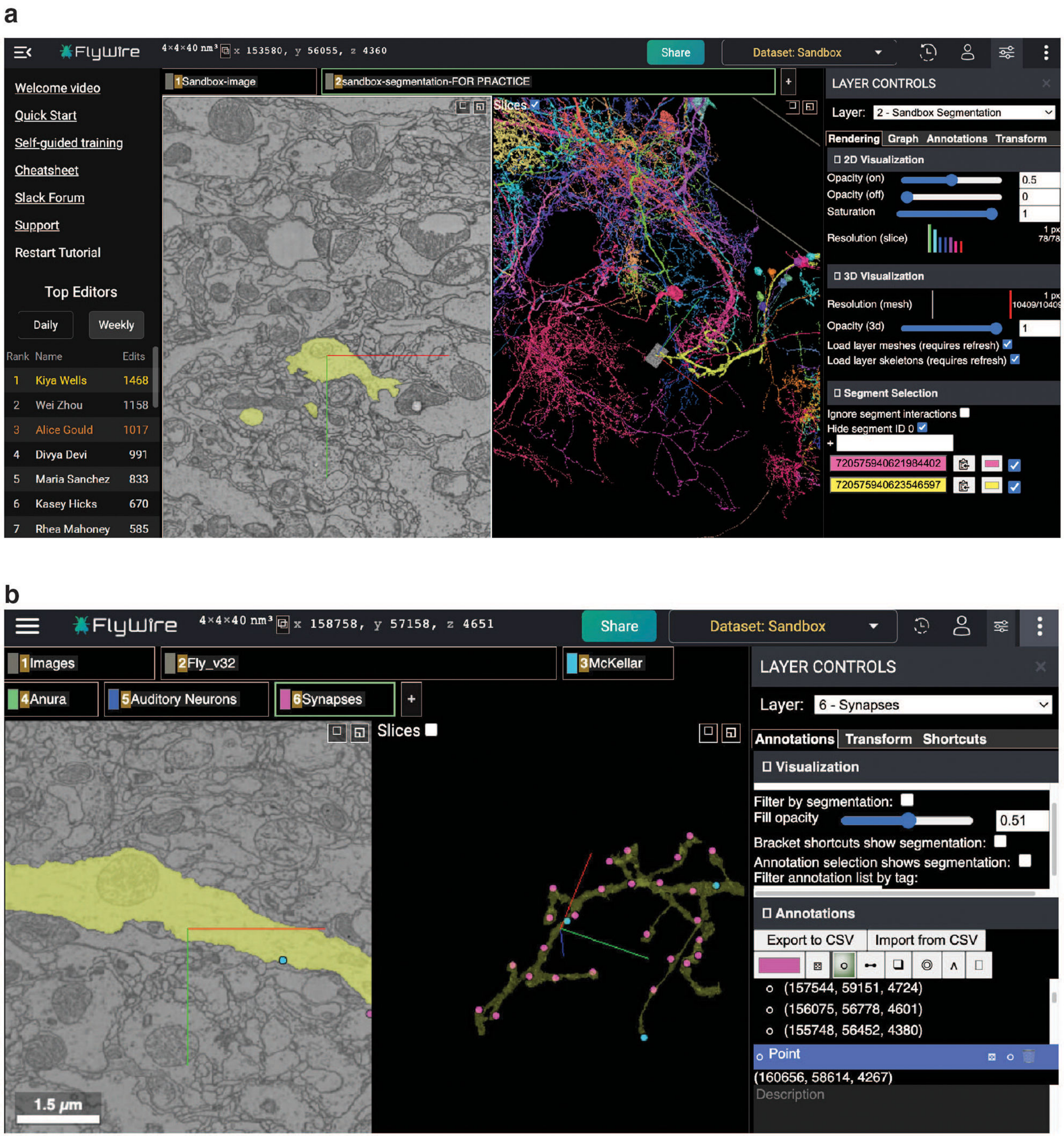
**Extended Data Fig. 3: Chunking the dataset.**

**(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. All 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



4×4×40 nm³ x 153580, y 56055, z 4360

Welcome video
Quick Start
Self-guided training
Cheatsheet
Slack Forum
Support
Restart Tutorial

Top Editors

Daily Weekly

| Rank | Name | Edits |
|---|---|---|
| 1 | Kiya Wells | 1468 |
| 2 | Wei Zhou | 1158 |
| 3 | Alice Gould | 1017 |
| 4 | Divya Devi | 991 |
| 5 | Maria Sanchez | 833 |
| 6 | Kasey Hicks | 670 |
| 7 | Rhea Mahoney | 585 |

1 Sandbox-image
2 sandbox-segmentation-FOR PRACTICE
Slices ☑

LAYER CONTROLS
Layer: 2 - Sandbox Segmentation
Rendering Graph Annotations Transform

□ 2D Visualization
Opacity (on) 0.5
Opacity (off) 0
Saturation 1
Resolution (slice) 1 px 78/78

□ 3D Visualization
Resolution (mesh) 1 px 10409/10409
Opacity (3d) 1
Load layer meshes (requires refresh) ☑
Load layer skeletons (requires refresh) ☑

□ Segment Selection
Ignore segment interactions □
Hide segment ID 0 ☑
+
720575940621984402 ☑
720575940623546597 ☑

4×4×40 nm³ x 158758, y 57158, z 4651

1 Images
2 Fly_v32
3 McKellar
4 Anura
5 Auditory Neurons
6 Synapses
Slices □

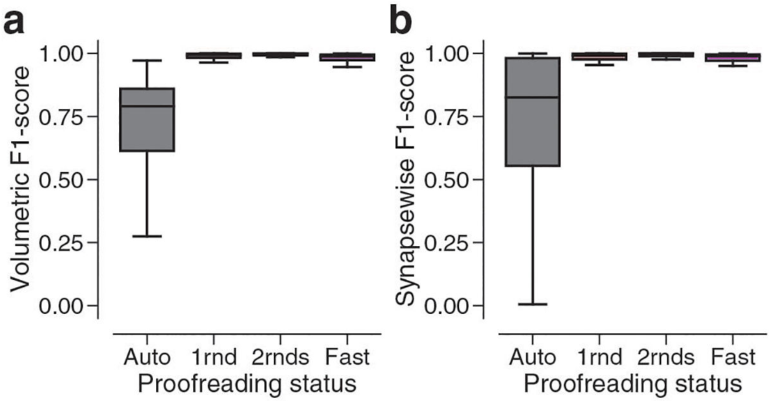
LAYER CONTROLS
Layer: 6 - Synapses
Annotations Transform Shortcuts

□ Visualization
Filter by segmentation: □
Fill opacity 0.51
Bracket shortcuts show segmentation: □
Annotation selection shows segmentation: □
Filter annotation list by tag:

□ Annotations
Export to CSV Import from CSV
○ (157544, 59151, 4724)
○ (156075, 56778, 4601)
○ (155748, 56452, 4380)
○ Point
(160656, 58614, 4267)
Description

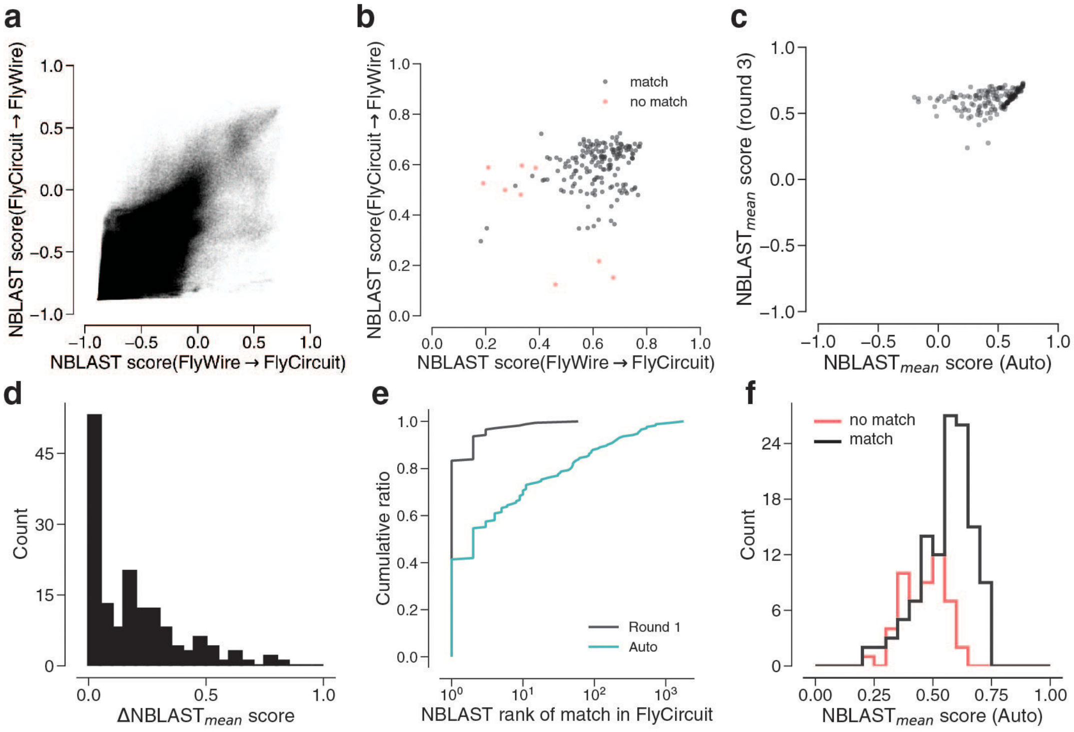
**Extended Data Fig. 5: The FlyWire proofreading platform.**

**(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.



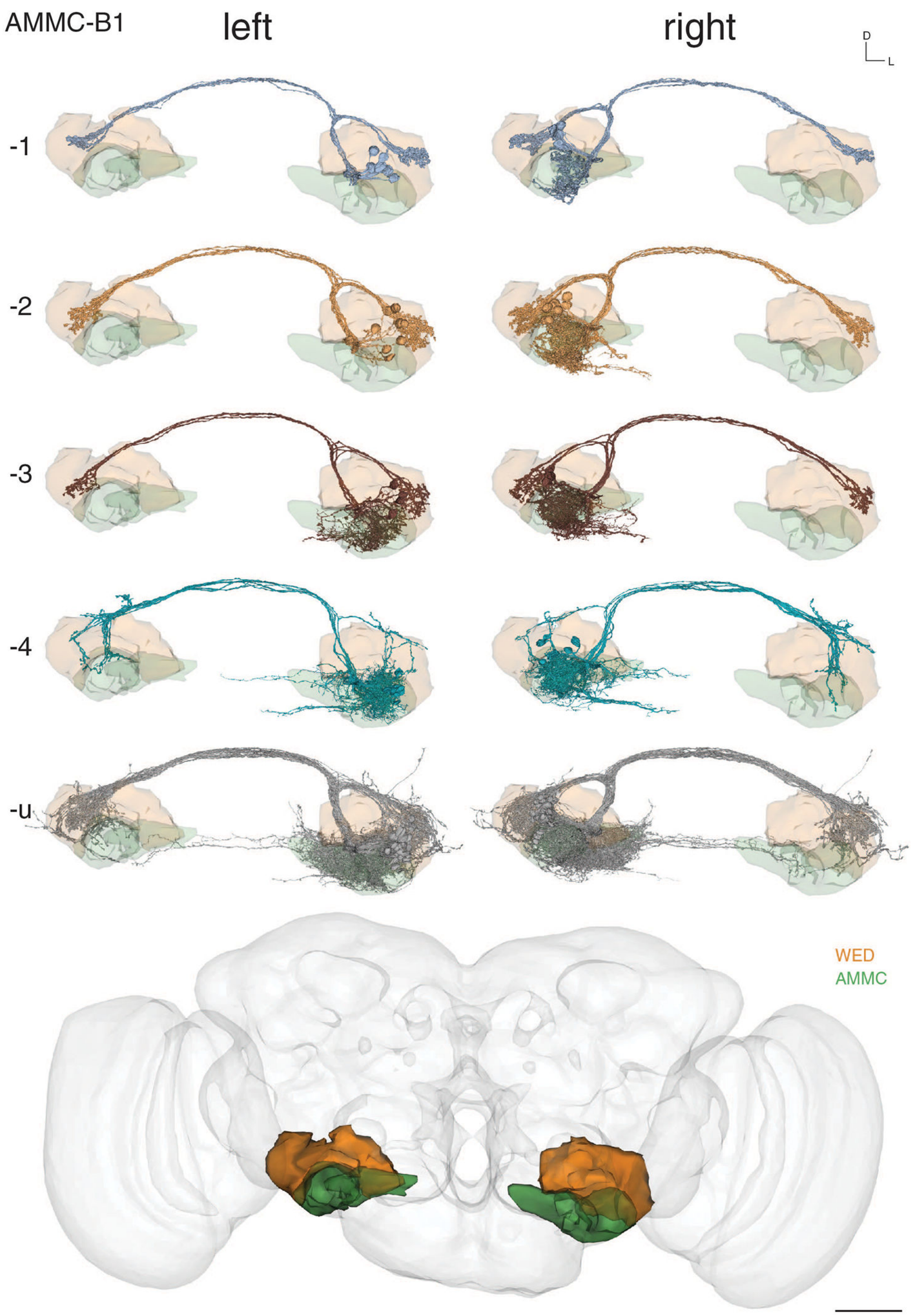
**Extended Data Fig. 6: Fast proofreading in FlyWire.**

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.



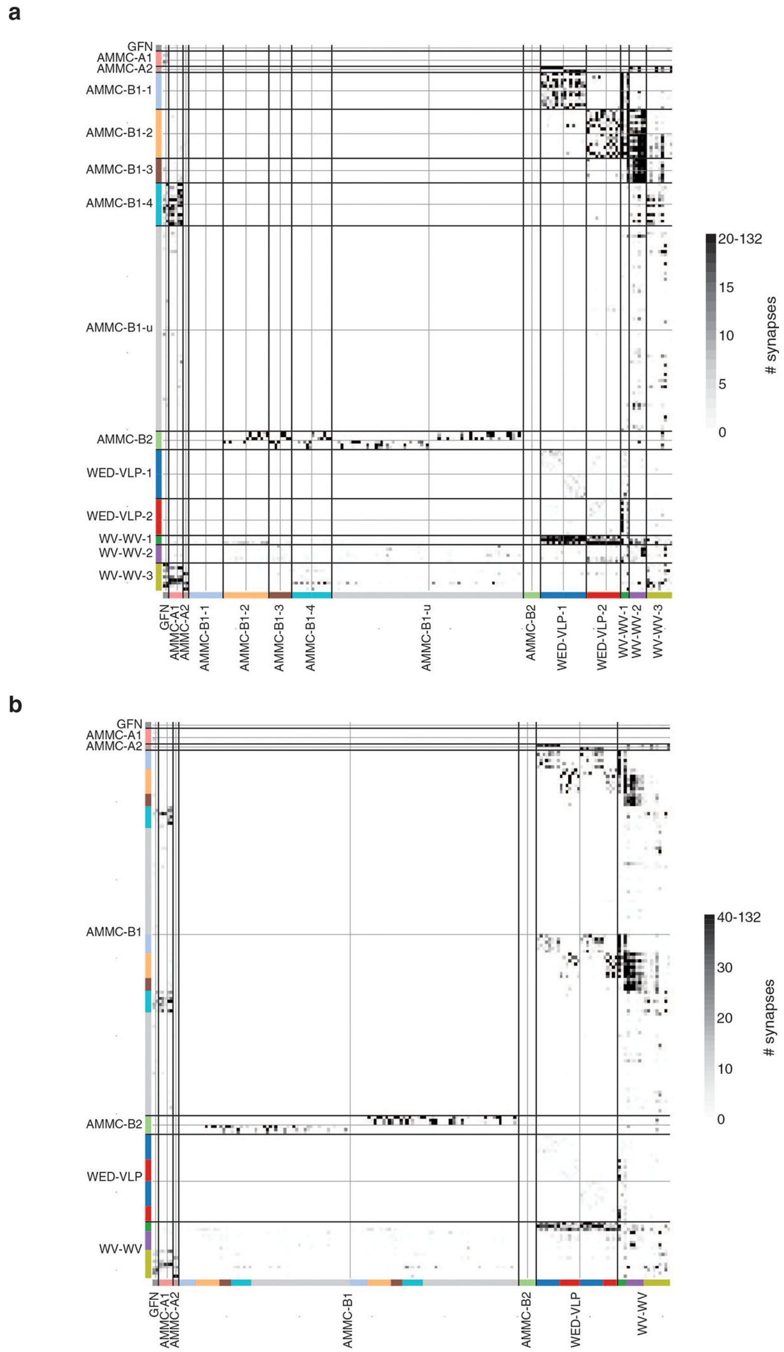
**Extended Data Fig. 7: NBLAST based analysis of segmentation accuracy.**

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 Figure 6b reordered by putative subtype **(b)** Same diagram as in Figure 6b with different colormap threshold.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## ACKNOWLEDGMENTS

We acknowledge support from NIH BRAIN Initiative RF1 MH117815 to HSS and MM. MM further received funding through an HHMI Faculty Scholar award and an NIH R35 Research Program Award. HSS further received NIH funding through RF1MH123400, U01MH117072, U01MH114824. HSS further 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 by Stephan Saalfeld, Eric Trautman, and Davi Bock. We are grateful to Davi Bock and Zhihao Zheng for discussions about FAFB. We thank Greg Jefferis, Davi Bock, Albert Cardona, Andrew Seeds, Steffi Hampel and Rachel Wilson for advice regarding the community. We thank Greg Jefferis and Philipp Schlegel (both with MRC Laboratory of Molecular Biology, Cambridge and University of Cambridge, Cambridge) for help with the brain renderings, transformations to FlyCircuit and NBLAST comparison with FlyCircuit neurons. We thank Garrett McGrath for computer system administration, and May Husseini for project administration. We are grateful to J. Maitin-Shepard for Neuroglancer. We are grateful to J. Buhmann and J. Funke for discussions about their synapse resource. We thank Nuno da Costa, Agnes Bodor, Celia David, and the Eyewire team for feedback on the proofreading system. We thank the Allen Institute for Brain Science founder, Paul G. Allen, for his vision, encouragement and support.

This work was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) contract number D16PC0005 to HSS. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

## REFERENCES

1. Ahrens MB, Orger MB, Robson DN, Li JM & Keller PJ Whole-brain functional imaging at cellular resolution using light-sheet microscopy. Nat. Methods 10, 413–420 (2013) doi:10.1038/nmeth.2434. [PubMed: 23524393]
2. White JG, Southgate E, Thomson JN & 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) doi:10.1098/rstb.1986.0056. [PubMed: 22462104]
3. Cook SJ et al. Whole-animal connectomes of both Caenorhabditis elegans sexes. Nature 571, 63–71 (2019) doi:10.1038/s41586-019-1352-7. [PubMed: 31270481]
4. Scheffer LK et al. A connectome and analysis of the adult Drosophila central brain. Elife 9, (2020) doi:10.7554/eLife.57443.
5. Coen P et al. Dynamic sensory cues shape song structure in Drosophila. Nature 507, 233–237 (2014) doi:10.1038/nature13131. [PubMed: 24598544]
6. Duistermars BJ, Pfeiffer BD, Hoopfer ED & Anderson DJ A Brain Module for Scalable Control of Complex, Multi-motor Threat Displays. Neuron 100, 1474–1490.e4 (2018) doi:10.1016/j.neuron.2018.10.027. [PubMed: 30415997]
7. Seelig JD & Jayaraman V Neural dynamics for landmark orientation and angular path integration. Nature 521, 186–191 (2015) doi:10.1038/nature14446. [PubMed: 25971509]
8. DasGupta S, Ferreira CH & Miesenböck G FoxP influences the speed and accuracy of a perceptual decision in Drosophila. Science 344, 901–904 (2014) doi:10.1126/science.1252114. [PubMed: 24855268]
9. Owald D et al. Activity of defined mushroom body output neurons underlies learned olfactory behavior in Drosophila. Neuron 86, 417–427 (2015) doi:10.1016/j.neuron.2015.03.025. [PubMed: 25864636]
10. Zheng Z et al. A Complete Electron Microscopy Volume of the Brain of Adult Drosophila melanogaster. Cell 174, 730–743.e22 (2018) doi:10.1016/j.cell.2018.06.019. [PubMed: 30033368]
11. Kim JS et al. Space-time wiring specificity supports direction selectivity in the retina. Nature 509, 331–336 (2014) doi:10.1038/nature13240. [PubMed: 24805243]
12. Haehn D et al. Design and Evaluation of Interactive Proofreading Tools for Connectomics. IEEE Trans. Vis. Comput. Graph. 20, 2466–2475 (2014) doi:10.1109/TVCG.2014.2346371. [PubMed: 26356960]

13. Knowles-Barley S et al. RhoanaNet Pipeline: Dense Automatic Neural Annotation. arXiv [q-bio.NC] (2016) access at http://arxiv.org/abs/1611.06973.

14. Zhao T, Olbris DJ, Yu Y & Plaza SM NeuTu: Software for Collaborative, Large-Scale, Segmentation-Based Connectome Reconstruction. Front. Neural Circuits 12, 101 (2018) doi:10.3389/fncir.2018.00101. [PubMed: 30483068]

15. Felsenberg J et al. Integration of Parallel Opposing Memories Underlies Memory Extinction. Cell 175, 709–722.e15 (2018) doi:10.1016/j.cell.2018.08.021. [PubMed: 30245010]

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.e8 (2018) doi:10.1016/j.neuron.2018.08.037. [PubMed: 30244885]

17. Zheng Z et al. Structured sampling of olfactory input by the fly mushroom body. bioRxiv (2020) doi:10.1101/2020.04.17.047167.

18. Li PH et al. Automated Reconstruction of a Serial-Section EM Drosophila Brain with Flood-Filling Networks and Local Realignment. bioRxiv 605634 (2019) doi:10.1101/605634.

19. Mitchell E, Keselj S, Popovych S, Buniatyan D & Sebastian Seung H Siamese Encoding and Alignment by Multiscale Learning with Self-Supervision. arXiv [cs.CV] (2019) access at http://arxiv.org/abs/1904.02643.

20. Deutsch D et al. The neural basis for a persistent internal state in Drosophila females. Elife 9, (2020) doi:10.7554/eLife.59502.

21. Baker CA, McKellar C, Nern A & Dorkenwald S Neural network organization for courtship song feature detection in Drosophila. bioRxiv (2020) doi:10.1101/2020.10.08.332148.

22. Pézier AP, Jezzini SH, Bacon JP & Blagburn JM Shaking B Mediates Synaptic Coupling between Auditory Sensory Neurons and the Giant Fiber of Drosophila melanogaster. PLoS One 11, e0152211 (2016) doi:10.1371/journal.pone.0152211. [PubMed: 27043822]

23. 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) doi:10.1016/j.cub.2011.02.041. [PubMed: 21530256]

24. Lee K, Zung J, Li P, Jain V & Sebastian Seung H Superhuman Accuracy on the SNEMI3D Connectomics Challenge. arXiv [cs.CV] (2017) access at https://arxiv.org/abs/1706.00120.

25. Dorkenwald S, Turner NL, Macrina T, Lee K & Lu R Binary and analog variation of synapses between cortical pyramidal neurons. bioRxiv (2019) doi:10.1101/2019.12.29.890319.

26. Zlateski A & Sebastian Seung H Image Segmentation by Size-Dependent Single Linkage Clustering of a Watershed Basin Graph. arXiv [cs.CV] (2015) access at http://arxiv.org/abs/1505.00249.

27. Maitin-Shepard J et al. google/neuroglancer: (2021). doi:10.5281/zenodo.5573294.

28. Chang F et al. Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst. 26, 1–26 (2008) doi:10.1145/1365815.1365816.

29. Priedhorsky R et al. Creating, destroying, and restoring value in Wikipedia. in Proceedings of the 2007 international ACM conference on Supporting group work 259–268 (2007). doi:10.1145/1316624.1316663.

30. Buhmann J et al. Automatic detection of synaptic partners in a whole-brain Drosophila electron microscopy data set. Nat. Methods 18, 771–774 (2021) doi:10.1038/s41592-021-01183-7. [PubMed: 34168373]

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). doi:10.1007/978-3-030-00934-2_36.

32. Staffler B et al. SynEM, automated synapse detection for connectomics. Elife 6, (2017) doi:10.7554/eLife.26414.

33. Dorkenwald S et al. Automated synaptic connectivity inference for volume electron microscopy. Nat. Methods (2017) doi:10.1038/nmeth.4206.

34. Huang GB, Scheffer LK & Plaza SM Fully-Automatic Synapse Prediction and Validation on a Large Data Set. Front. Neural Circuits 12, 87 (2018) doi:10.3389/fncir.2018.00087. [PubMed: 30420797]

35. Turner N et al. Synaptic Partner Assignment Using Attentional Voxel Association Networks. arXiv [cs.CV] (2019) access at http://arxiv.org/abs/1904.09947.

36. Buhmann J et al. Synaptic Partner Prediction from Point Annotations in Insect Brains. in Medical Image Computing and Computer Assisted Intervention – MICCAI 2018 309–316 (Springer International Publishing, 2018). doi:10.1007/978-3-030-00934-2_35.

37. Kreshuk A, Funke J, Cardona A & Hamprecht FA Who Is Talking to Whom: Synaptic Partner Detection in Anisotropic Volumes of Insect Brain. in Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015 661–668 (Springer International Publishing, 2015). doi:10.1007/978–3-319–24553-9_81.

38. Schneider-Mizell CM et al. Quantitative neuroanatomy for connectomics in Drosophila. Elife 5, (2016) doi:10.7554/eLife.12059.

39. Takemura S-Y et al. A visual motion detection circuit suggested by Drosophila connectomics. Nature 500, 175–181 (2013) doi:10.1038/nature12450. [PubMed: 23925240]

40. Meinertzhagen IA Of what use is connectomics? A personal perspective on the Drosophila connectome. J. Exp. Biol. 221, (2018) doi:10.1242/jeb.164954.

41. Chiang A-S et al. Three-dimensional reconstruction of brain-wide wiring networks in Drosophila at single-cell resolution. Curr. Biol. 21, 1–11 (2011) doi:10.1016/j.cub.2010.11.056. [PubMed: 21129968]

42. Costa M, Manton JD, Ostrovsky AD, Prohaska S & Jefferis GSXE NBLAST: Rapid, Sensitive Comparison of Neuronal Structure and Construction of Neuron Family Databases. Neuron 91, 293–311 (2016) doi:10.1016/j.neuron.2016.06.012. [PubMed: 27373836]

43. Tootoonian S, Coen P, Kawai R & Murthy M Neural representations of courtship song in the Drosophila brain. J. Neurosci. 32, 787–798 (2012) doi:10.1523/JNEUROSCI.5104-11.2012. [PubMed: 22262877]

44. Lai JS-Y, Lo S-J, Dickson BJ & Chiang A-S Auditory circuit in the Drosophila brain. Proc. Natl. Acad. Sci. U. S. A. 109, 2607–2612 (2012) doi:10.1073/pnas.1117307109. [PubMed: 22308412]

45. Vaughan AG, Zhou C, Manoli DS & Baker BS Neural pathways for the detection and discrimination of conspecific song in D. melanogaster. Curr. Biol. 24, 1039–1049 (2014) doi:10.1016/j.cub.2014.03.048. [PubMed: 24794294]

46. Yamada D et al. GABAergic Local Interneurons Shape Female Fruit Fly Response to Mating Songs. J. Neurosci. 38, 4329–4347 (2018) doi:10.1523/JNEUROSCI.3644-17.2018. [PubMed: 29691331]

47. Kamikouchi A et al. The neural basis of Drosophila gravity-sensing and hearing. Nature 458, 165–171 (2009) doi:10.1038/nature07810. [PubMed: 19279630]

48. Patella P & Wilson RI Functional Maps of Mechanosensory Features in the Drosophila Brain. Curr. Biol. 28, 1189–1203.e5 (2018) doi:10.1016/j.cub.2018.02.074. [PubMed: 29657118]

49. 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) doi:10.1002/cne.24877. [PubMed: 32012264]

50. Clemens J et al. Connecting Neural Codes with Behavior in the Auditory System of Drosophila. Neuron 87, 1332–1343 (2015) doi:10.1016/j.neuron.2015.08.014. [PubMed: 26365767]

51. Azevedo AW & Wilson RI Active Mechanisms of Vibration Encoding and Frequency Filtering in Central Mechanosensory Neurons. Neuron 96, 446–460.e9 (2017) doi:10.1016/j.neuron.2017.09.004. [PubMed: 28943231]

52. von Reyn CR et al. A spike-timing mechanism for action selection. Nat. Neurosci. 17, 962–970 (2014) doi:10.1038/nn.3741. [PubMed: 24908103]

53. Allen MJ, Godenschwege TA, Tanouye MA & Phelan P Making an escape: development and function of the Drosophila giant fibre system. Semin. Cell Dev. Biol. 17, 31–41 (2006) doi:10.1016/j.semcdb.2005.11.011. [PubMed: 16378740]

54. Phelan P et al. Molecular mechanism of rectification at identified electrical synapses in the Drosophila giant fiber system. Curr. Biol. 18, 1955–1960 (2008) doi:10.1016/j.cub.2008.10.067. [PubMed: 19084406]

55. Morley EL, 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) doi:10.1242/jeb.068940. [PubMed: 22723479]

56. Giles J Internet encyclopaedias go head to head. Nature 438, 900–901 (2005) doi:10.1038/438900a. [PubMed: 16355180]

57. Zung J, Tartavull I, Lee K & Seung HS An Error Detection and Correction Framework for Connectomics. in Advances in Neural Information Processing Systems 30 (eds. Guyon I et al.) 6818–6829 (Curran Associates, Inc., 2017). access at http://papers.nips.cc/paper/7258-an-error-detection-and-correction-framework-for-connectomics.pdf.

58. Costa M, Schlegel P & Jefferis G FlyCircuit Dotprops. (2016). doi:10.5281/zenodo.5205616.

59. Bates AS et al. The natverse, a versatile toolbox for combining and analysing neuroanatomical data. eLife vol. 9 (2020) doi:10.7554/elife.53350.

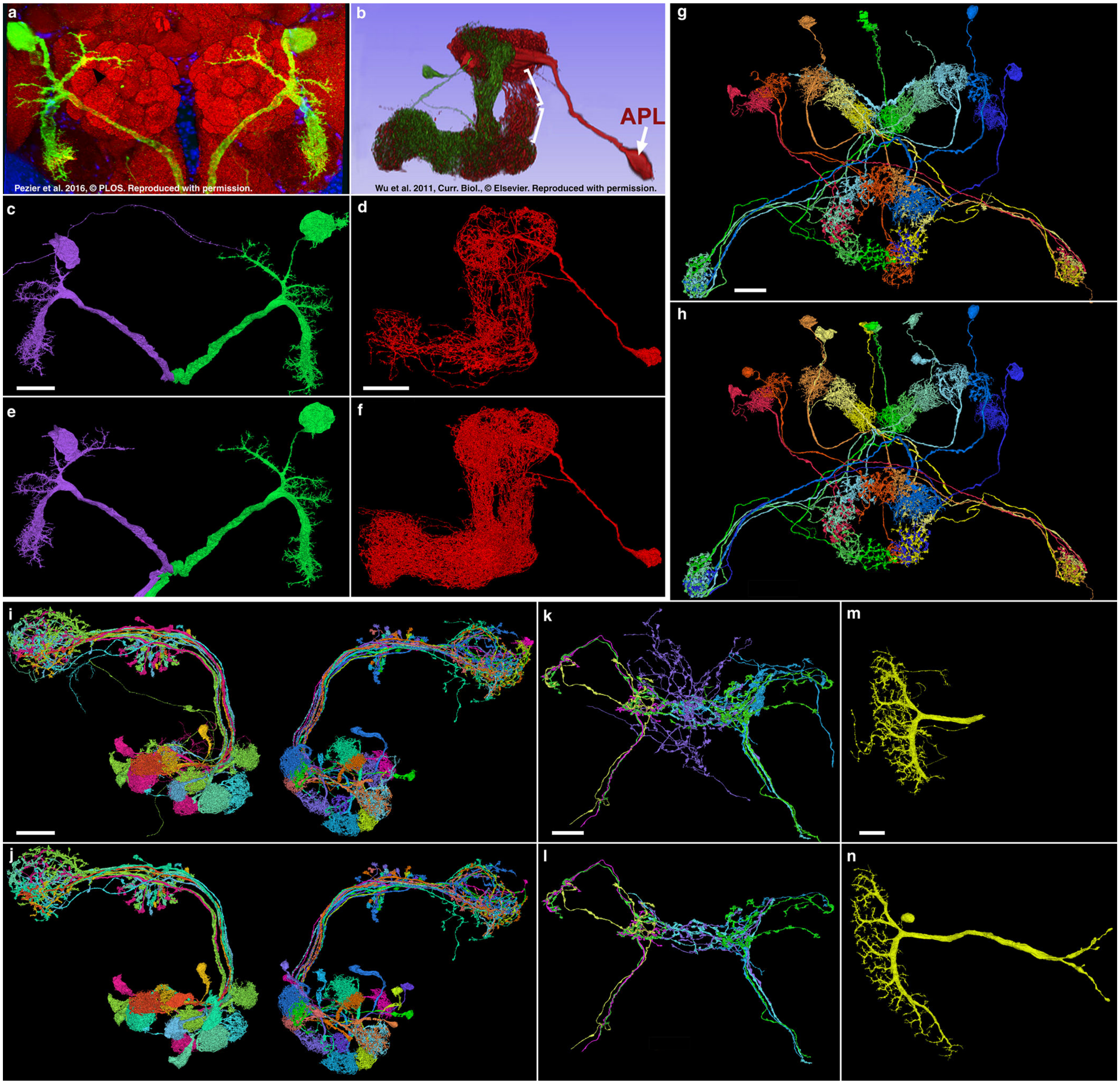

a
Pezier et al. 2016, © PLOS. Reproduced with permission.
b
APL
Wu et al. 2011, Curr. Biol., © Elsevier. Reproduced with permission.
c
d
e
f
g
h
i
j
k
l
m
n

**Figure 1. Assessing segmentation quality using known neurons.**

(**a-d**) Comparison of light microscopy-level stains of giant fiber neurons [22] (**a**) and a mushroom body APL neuron [23](**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 frontal except APL and central complex neurons: dorso-frontal view. Scale bars: (c, d, e, f, i, j) 30 µm; (g, h, k, l) 15 µm; (m, n) 20 µm.

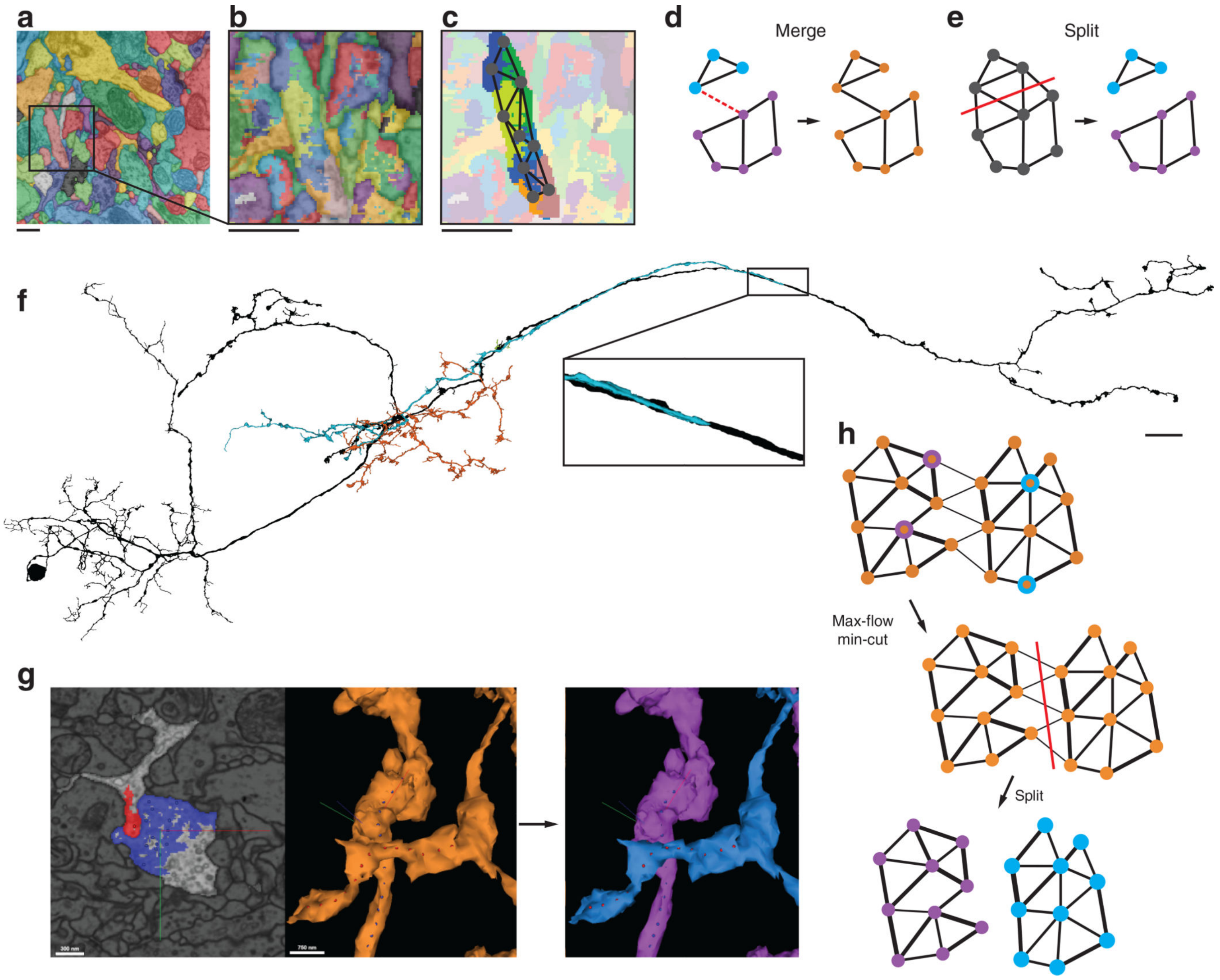


**Figure 2. Proofreading the supervoxel graph.**

**(a)** Automated segmentation overlaid on the EM data. Each color represents an individual putative cell. **(b)** Different colors represent the supervoxels that make up the 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, purple). **(f)** Example neuron after proofreading (black). Green, blue and red components were removed during proofreading. While edit operations have global effects, the edits to the supervoxel graph themselves are performed at a local level. **(g)** For splits, users place points (red and blue dots) either in 2D (left) or 3D (center panel) that are linked to the underlying supervoxels (left panel). The proofreading backend then automatically determines which edges need to be removed and performs the split (right panel). The panels are screenshots from FlyWire’s neuroglancer.

The colored lines represent coordinate axes: red (x), green (y), blue (z). **(h)** For the operation shown in (g) the backend performs max-flow min-cut on the local supervoxel graph to determine the optimal cut that separates the user-defined input locations (blue and purple framed circles). The thickness of the edges symbolizes the edge weight (cartoon). Scale bars (a,b,c): 1μm; (f): 10 μm

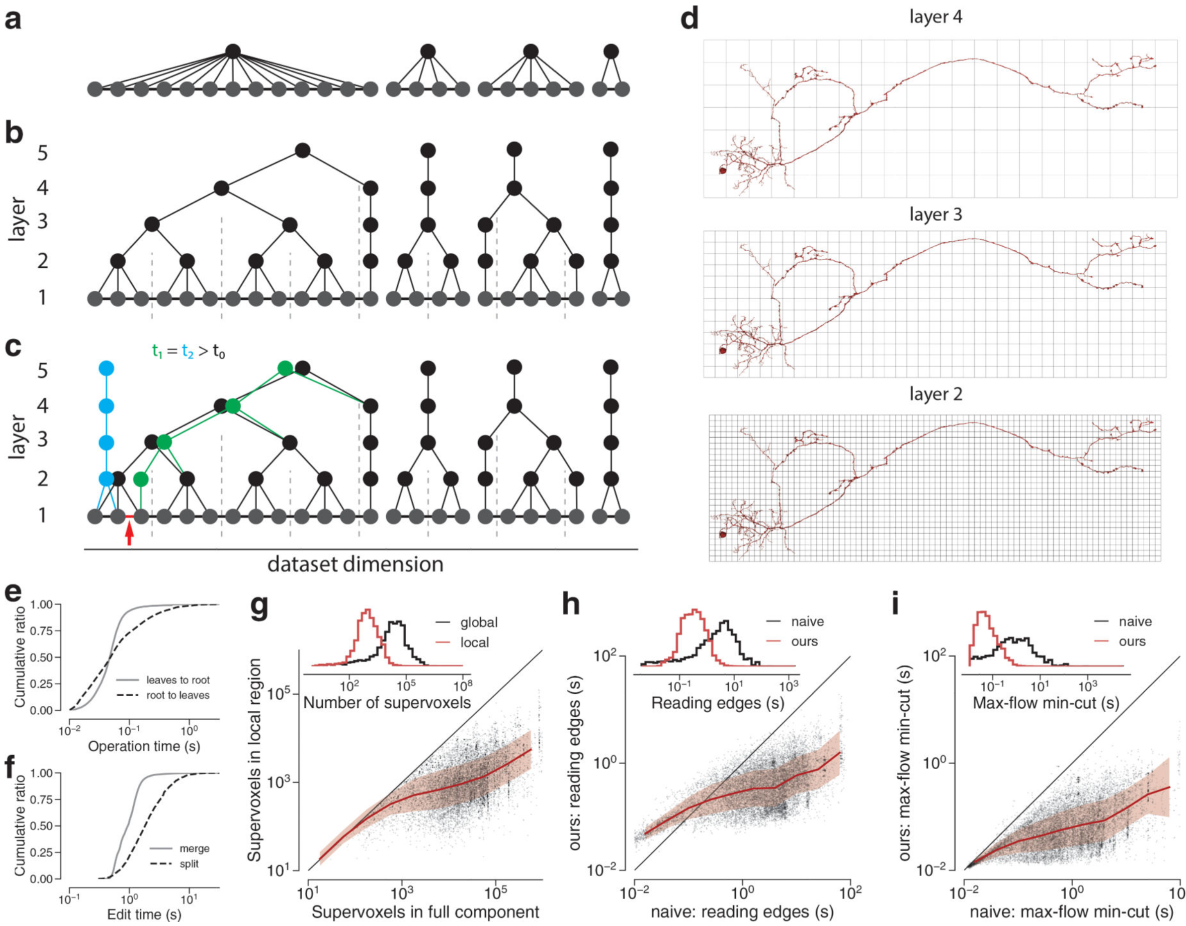


**Figure 3. The ChunkedGraph approach for proofreading supervoxel graphs.**

**(a)** One-dimensional representation of the supervoxels graph. In the simplest approach (naive), connected component information (neuronal component) is stored in a dedicated parent node. **(b)** In an alternative data structure 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. **(c)** Illustration of how edits to the ChunkedGraph (here, a split; indicated by the red arrow and removed red edge) affect the supervoxel graph to recompute the neuronal connected components. **(d)** Chunk size (represented by the grid) along each dimension in different layers. **(e)** Server response times for the remapping of the connected components from root to supervoxel (N=3,080,494) and supervoxel to root (N=12,096) **(f)** as well as splits (N=2,497) and merges (N=4,612) for real user interactions in the beta-phase of FlyWire. **(g)** Number of supervoxels that need to be loaded for a split (global vs local) **(h, i)** Reading speed **(h)** and speed of max-flow min-cut calculations **(i)** for the ChunkedGraph and a naïve approach. The red lines in (g, h, i) are mean and the shaded area standard deviation of bins along x-axis (10 bins); N=15,233 split operations. N in (e, f) are the number of observed requests to the server.

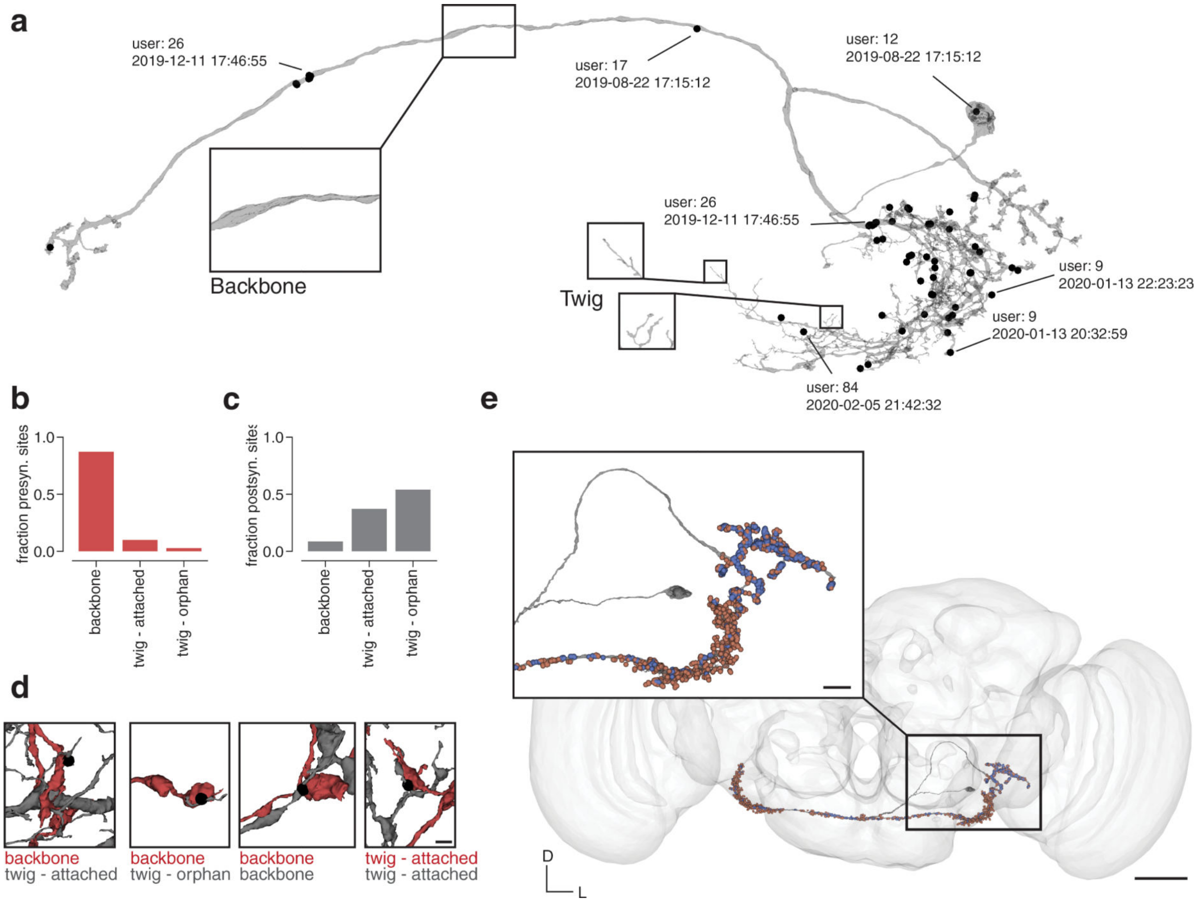


**Figure 4. Attaching automatically detected synapses to neurons.**

**(a)** Each edit (black dot) is linked to a user and timestamp enabling the retrieval of the edit history and credit assignment post-hoc. **(b, c)** Classification of pre- **(b)** and post-synaptic **(c)** segments based on their morphology and whether they are attached to a bigger component that will be attached during a conservative procedure. **(d)** Examples of these assessments. **(e)** AMMC-A2 neuron (left) with automatically detected synapses displayed as balls (blue: presynaptic (N=5140), red: postsynaptic (N=1669), balls overlap). Scale bar: (d): 1 μm; (e) 50 μm, (e, inset): 10 μm

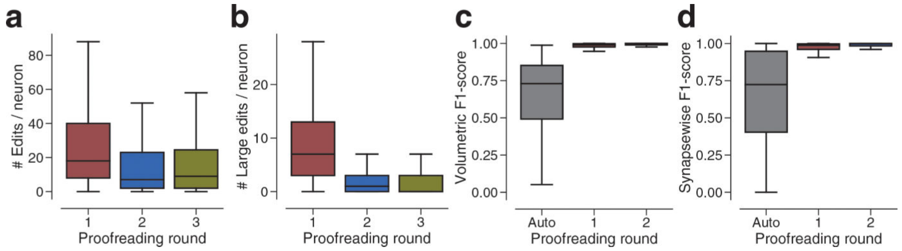


**Figure 5. Proofreading in FlyWire.**

Analysis of 183 triple-proofread neurons **(a)** Number of edits per neuron and proofreading round (medians: 1: 18, 2: 7, 3: 9, means: 1: 36.5, 2: 18.0, 3: 25.7). **(b)** Number of edits per neuron and proofreading round restricted to large edits (> 1μm$^3$, medians: 1: 7, 2: 1, 3: 0, means: 1: 10.9, 2: 2.5, 3: 2.6). **(c, d)** F1-Scores (0–1, higher is better; with respect to proofreading results after three rounds) between different proofreading rounds according to volumetric completeness **(c)** (medians: Auto: 0.730, 1: 0.989, 2: 0.999, means: Auto: 0.665, 1: 0.968, 2: 0.984) and assigned synapses **(d)** (medians: Auto: 0.724, 1: 0.988, 2: 0.998, means: Auto: 0.642, 1: 0.942, 2: 0.970). “Auto” refers to reconstructions without proofreading. Boxes are interquartile ranges (IQR), whiskers are set at 1.5 x IQR.

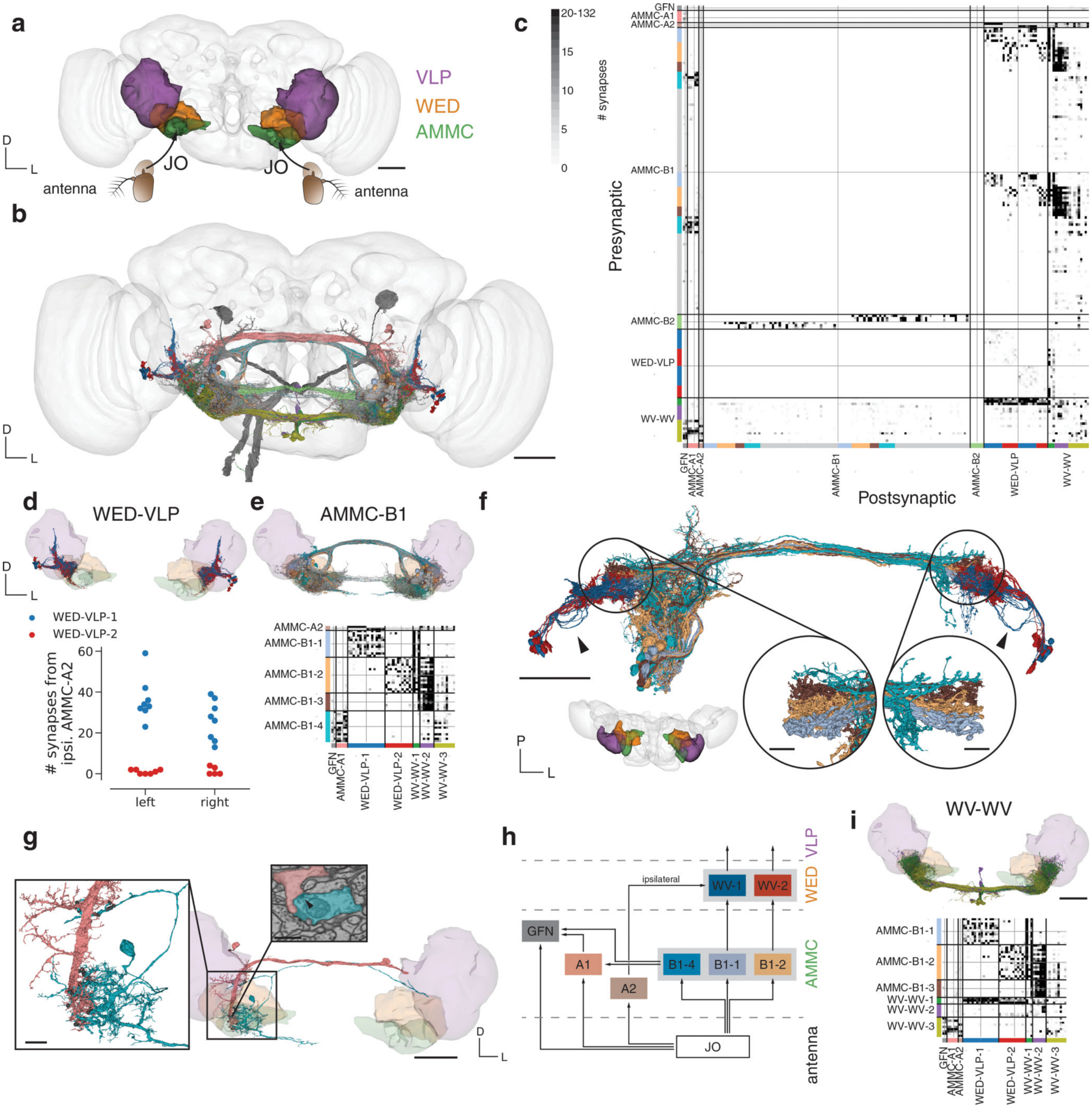


**Figure 6. Connectivity between mechanosensory neurons extracted with FlyWire.**

**(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 (Johnston's Organ) 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/top: left hemisphere, right/bottom: right hemisphere) and colored bars separate putative cell types within each cell class. **(d)** WED-VLP type 1 and 2 neurons, separated based on 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 AMMCA1 neurons, and their connectivity matrix. Scale bars: 50 μm, insets in (f), (g): 10 μm, EM inset in (g): 500 nm