NeuVue: A Framework and Workflows for High-Throughput Electron Microscopy Connectomics Proofreading

Daniel Xenes[a], Lindsey M. Kitchell[a], Patricia K. Rivlin[a], Rachel Brodsky[a], Hannah Gooden[a], Justin Joyce[a], Diego Luna[a], Raphael Norman-Tenazas[a], Devin Ramsden[a], Kevin Romero[a], Victoria Rose[a], Marisel Villafane-Delgado[a], William Gray-Roncal[a], Brock Wester[a]

[a] Research & Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory

Funding:

This work was supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2017-17032700004 under the MICrONS program. 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 the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation therein.

1. Abstract

NeuVue is a software platform created for large-scale proofreading of machine segmentation and neural circuit reconstruction in high-resolution electron microscopy connectomics datasets. The NeuVue platform provides a robust web-based interface for proofreaders to collaboratively view, annotate, and edit segmentation and connectivity data. A backend queuing service organizes proofreader tasks into purpose-driven task types and increases proofreader throughput by limiting proofreader actions to simple, atomic operations. A collection of analytical dashboards, data visualization tools, and Application Program Interface (API) capabilities provide stakeholders real-time access to proofreading progress at an individual proofreader level as well as insights on task generation priorities. NeuVue is agnostic to the underlying data being proofread and improves upon the traditional proofreader experience through quality-of-life features that streamline complex editing operations such as splitting and merging objects in dense nanoscale segmentation.

NeuVue heavily leverages cloud resources to enable proofreaders to simultaneously access and edit data on the platform. Production-quality features such as load-balancing, auto-scaling, and pre-deployment testing are all integrated into the platform's cloud architecture. Additionally, NeuVue is powered by well-supported open-source connectomics tools from the community such as Neuroglancer, PyChunkedGraph, and Connectomics Annotation Versioning Engine (CAVE). The modular design of NeuVue facilitates easy integration and adoption of useful community tools to allow proofreaders to take advantage of the latest improvements in data visualization, processing, and analysis.

We demonstrate our framework through proofreading of the mouse visual cortex data generated on the IARPA MICrONS Project. This effort has yielded over 40,000 proofreader edits across the 2 petavoxels of "Minnie" neuroimaging data. 44 unique proofreaders of various skill levels have logged a cumulative 3,740 proofreading hours, and we have been able to validate the improved connectivity of thousands of neurons in the volume. With sustained development on the platform, new integrated error detection and error correction capabilities, and continuous improvements to the proofreader model, we believe that the NeuVue framework can enable high-throughput proofreading for large-scale connectomics datasets of the future.

2. Introduction

As high-resolution electron microscopy (EM) data acquisition capabilities continue to advance, it has become common for research labs to image increasingly large (e.g., > 100 teravoxels) data volumes (Shapson-Coe et al. 2021; MICrONS Consortium et al. 2021; Turner et al. 2022; Dorkenwald et al. 2019). Following image volume collection, automated methods are applied to segment and identify objects of interest, such as neurons, nuclei, and synapses (Januszewski et al. 2018; Popovych et al. 2022; Yin et al. 2020). The performance of these algorithms has dramatically improved in the past few years; however, current methods are still imperfect and require significant human intervention (Scheffer et al. 2020; Dorkenwald et al. 2022). An end goal of these efforts, for which accurate segmentation is crucial, is to construct a connectome of sufficient quality for research into the underlying connectivity of the brain. In an ideal environment, expert proofreaders would visually inspect each segmentation result and connection point (i.e., synapse). As data volumes increase in size, manual inspection becomes intractable and so careful choices need to be made, balancing and prioritizing corrections which

maximize impact on connectome quality, based on available manpower (Plaza et al. 2014, Hubbard et al. 2020).

Several existing frameworks for proofreading connectomics data exist, including CATMAID (Saalfeld et al. 2009), Eyewire (Kim et al. 2014), Flywire (Dorkenwald et al. 2022), WebKnossos (Boergens et al. 2017), and NeuTu (Zhao et al. 2018). Our approach offers a complementary framework with an emphasis on atomic tasks; flexibility and scalability to accommodate a variable size userbase that includes both novices and experts; an open-source, data agnostic approach; and a web-based (cross-platform, zero install) solution that integrates easily with key tools in the community for error detection and correction. In this work, we share our key infrastructure and approach, and demonstrate sample workflows used to improve the MICrONS visual cortex dataset (MICrONS Consortium et al. 2021).

3. Materials and Methods

Below we describe NeuVue, a system developed for high-throughput connectomics proofreading. The high-level diagram of the proofreading workflow shown in Figure 1 illustrates how error identification methods and proofreading analytics interconnect with the NeuVue ecosystem to form a complete "proofreading workflow." This proofreading workflow begins with generating proofreading tasks through a multifaceted error identification analysis. The generated tasks are then queued to proofreaders who then complete the tasks through the NeuVue interface. During this process, proofreader actions are recorded to analyze their impact on the underlying data volume quality.

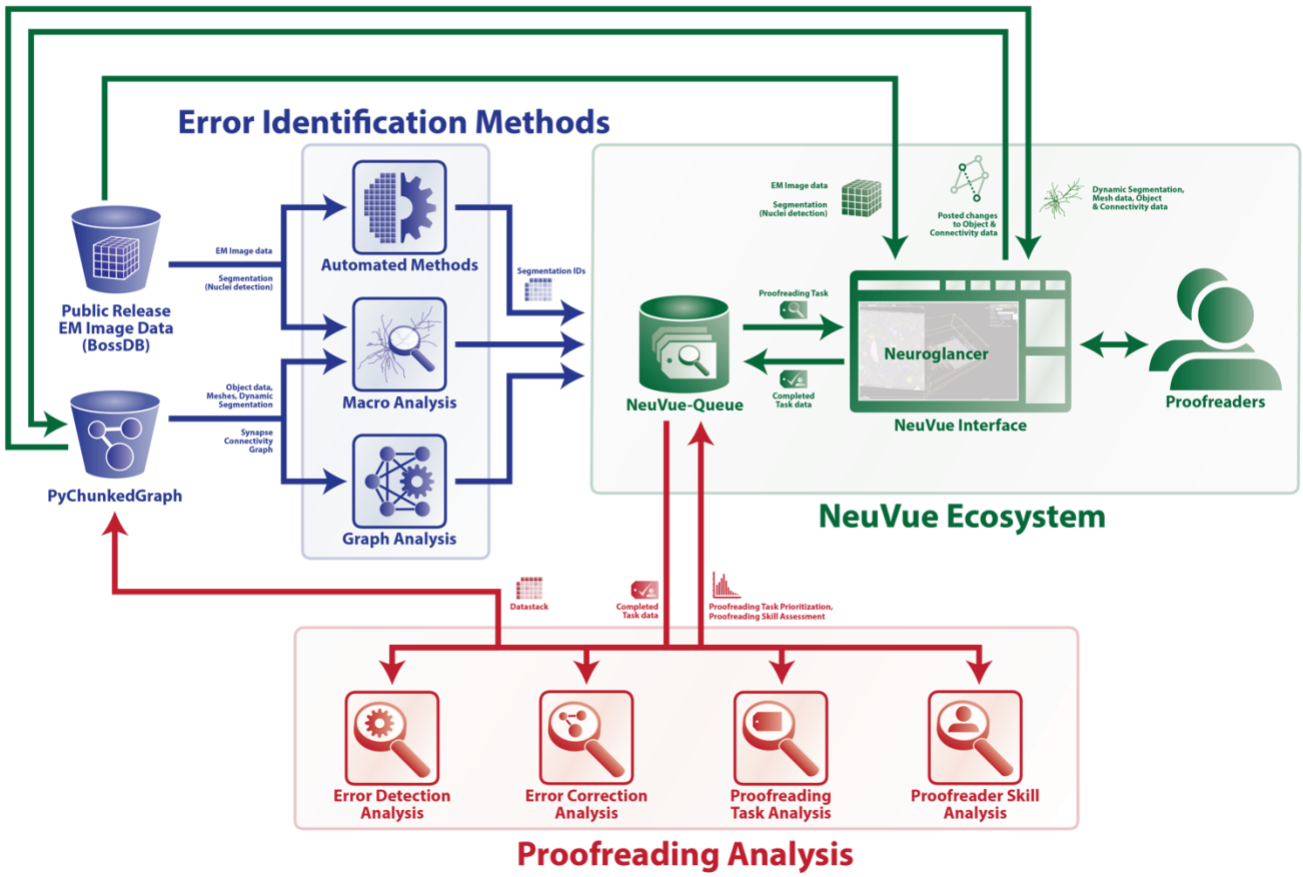

Figure 1. Diagram of the high-level workflow and connectivity of the various components of the current proofreading system. Error identification methods (blue) generate tasks that are then queued in the NeuVue ecosystem (green). Tasks are completed within this ecosystem and fed back to the original data sources. Proofreading Analysis (red) uses the completed task data to provide quantified improvements to the connectivity and skill assessment for individual proofreaders.

3.1 NeuVue Design

The NeuVue system is composed of the following components: a cloud-based, highly-available task queue service called NeuVue-Queue, a web browser-based proofreader interface with integrated community tools, and an analysis interface that provides information on available, pending, and completed proofreading tasks.

The NeuVue system is designed for flexibility of both proofreading task archetypes and data sources. We developed a simple API interface for NeuVue administrators to create atomic task types and assign tasks to proofreaders (Figure 2). The modular design of NeuVue permits connectivity with different image and segmentation data sources, but for the initial system design and demonstration, we point to data volumes and systems generated on the Intelligent Advanced Research Projects Activity (IARPA) Machine Intelligence from Cortical Networks (MICrONS) program.

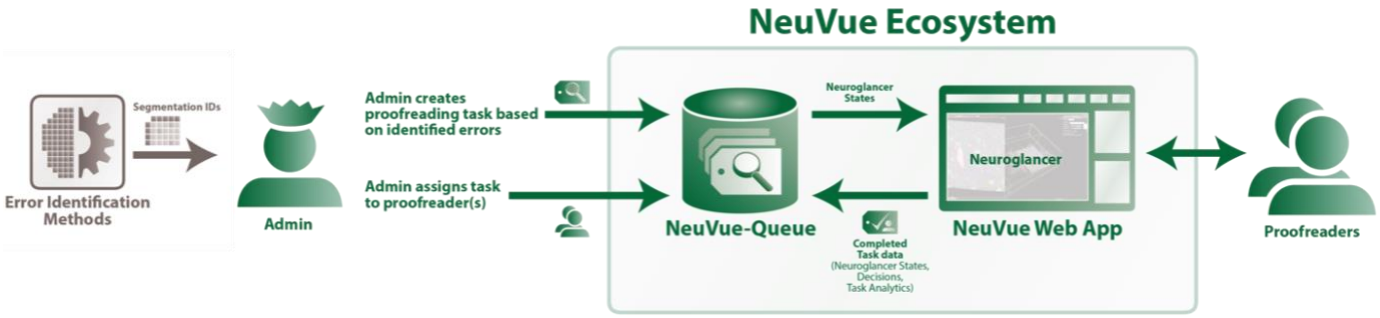

Figure 2. Diagram of the high-level design of the NeuVue proofreading workflow. Segmentation IDs sourced from error detection methods are provided to NeuVue administrators to create individual tasks. NeuVue administrators create the task type based on the identified error information and assign it to proofreaders through the NeuVue-Queue API. Proofreaders interact with the NeuVue Web App to view Neuroglancer states that complete the task, after which the app sends oover the completed task data back to NeuVue queue where it can be used for further analysis.

3.2 Connectomics Data Sources

We demonstrate the capabilities of the NeuVue system with publicly-released EM imagery of the mouse visual cortex from both the “Minnie” (MICrONS Consortium et al. 2021) and “Pinky” datasets (Dorkenwald et al. 2019) generated on the MICrONS Program. For these datasets, the associated fine-aligned imagery, static segmentation, and 3D meshes are stored in BossDB as precompute data (Hider et al. 2022, Maitin-Shepard 2020).

Additionally, we use a dynamic PyChunkedGraph (Dorkenwald et al. 2022) segmentation which is a direct descendent of the publicly-released MICrONS data. This dataset is directly linked to several annotation tables for soma and nuclei locations, cell typing, and connectivity which are automatically updated as edits are made by the proofreaders. To retrieve this dynamic segmentation, the NeuVue system connects to the Minnie datastack (“minnie65_phase3_v1”), (i.e., the collection of imagery,

dense segmentation, and annotation tables), through the PyChunkedGraph (Dorkenwald et al. 2022) and Connectomics Annotation Versioning Engine (CAVE) tools (*Seung-Lab/CAVEclient* [2018] 2022). The MICrONS Proofreading Coordination Group have also provided us a "sandbox" Pinky PyChunkedGraph that we use for training and testing purposes.

3.3 Proofreading Queue and Client Architecture

NeuVue-Queue provides a RESTful API designed to deliver tasks generated from analytical methods to the proofreaders in a priority queue system. NeuVue-Queue's backend consists of a MongoDB database coupled with an Express application server which is hosted on a scalable fleet of Docker containers. NeuVue-Queue and Neuvue-Client were developed as an extension of Colocard, a tool used for previous MICrONS evaluation efforts as a part of the CONFIRMS framework (Bishop et al. 2021).

To ensure that NeuVue-Queue can handle high proofreader loads mitigate the risk of data loss or service outages, we deployed NeuVue-Queue in an AWS cloud environment with token-based authentication, automated backups, and autoscaling. The queue stores all proofreading tasks under unique "namespaces" or task types. A proofreading "task" is represented as a standardized schema defining attributes of the task such as instructions, priority, assignee, author, Neuroglancer state, metadata, and telemetry of when the task was created, opened, and completed. Tasks are designed to be "stateless", where one task does not depend on other tasks and can be completed independently in any order. The "priority" attribute of the task allows administrators to discretely control the order in which tasks are completed by proofreaders.

NeuVue administrators can create tasks using the NeuVue-Client, a python package which interfaces with NeuVue-Queue API to query, modify, and post tasks to the queue. The client conforms with the numpy and pandas data standard to provide administrators a flexibility and autonomy when managing proofreader tasks for multiple proofreaders (Harris et al. 2020, Reback et al. 2022).

3.4 Proofreading Interface Architecture

The NeuVue proofreading interface is a web application built with the Django web framework that provides proofreaders the ability to view and complete their assigned tasks as well as a suite of administrator tools for managing queues and task types. It leverages a modified version of Google's Neuroglancer software (Maitin-Shepard 2020) to visualize the imagery and segmentation data through modern browsers. We designed the NeuVue web application to support an initial, predetermined set of task archetypes (e.g., splitting, merging, validating, training) while building in the ability to flexibly create new task schemas on demand in order to accommodate multifaceted proofreading goals as they evolve.

Proofreaders sign into the NeuVue interface ("NeuVue Web App") to work on queued proofreading tasks retrieved from NeuVue-Queue. The associated image and segmentation data for the task is shown in an embedded Neuroglancer window on their workspace page (Figure 5). Proofreaders can then complete the assigned tasks via several routes such as manual edits, forced-choice decisions, or generating annotations on spatial data. NeuVue also provides high-level administrator access for user management and analytics.

3.4.1 Accessing the Proofreading Interface

The proofreading interface webpage (https://app.neuvue.io) can be accessed via most web browsers following a Google OAuth login. Since the PyChunkedGraph dynamic segmentation can only be accessed through authorized Google accounts, we adopted a Google OAuth authentication system to provide seamless access to the app and data through one Google account. To give a proofreader access to NeuVue, an administrator must configure the appropriate permissions and assign the proofreader's account tasks, as outlined below. Following login, proofreaders are directed to the main homepage, which displays a welcome greeting, a panel for recent changes, and a navigation menu (Figure 3).

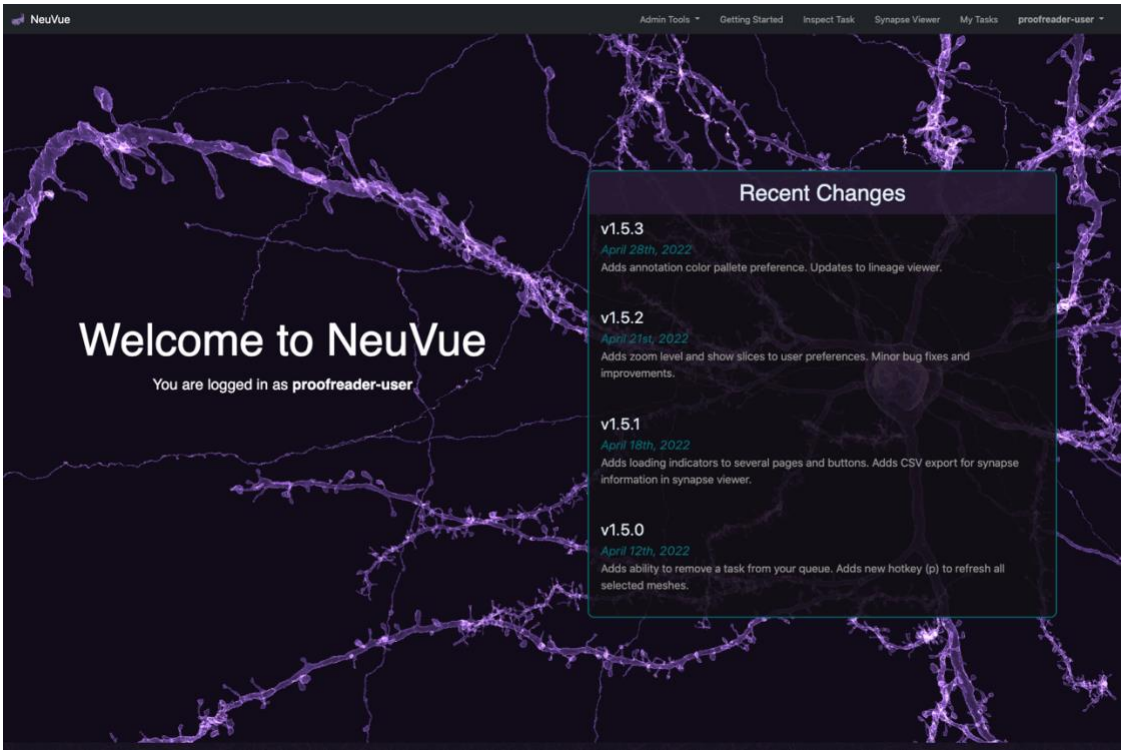

Figure 3. NeuVue landing page. This page is shown once a user with the appropriate access logs in with their Google OAuth account. A welcome greeting, panel for recent changes, and navigation bar are shown for quick access to other pages.

To access their assigned tasks, a proofreader can navigate to the "My Tasks" page (Figure 4). It contains tabularized information on open, pending, and closed tasks assigned to the proofreader as well as descriptive statistics on performance and completion speed. Users are not limited to working on one task type at a time and can have different task types open for completion in parallel that use different data sources. If a proofreader has pending or open tasks for one of the available task types, they can click the "Start Proofreading" button to enter the "Workspace" page and begin completing the tasks. Depending on a proofreader's level of experience with a task, proofreaders can also self-manage their queues by adding more tasks from an unassigned pool or removing tasks from their queue.

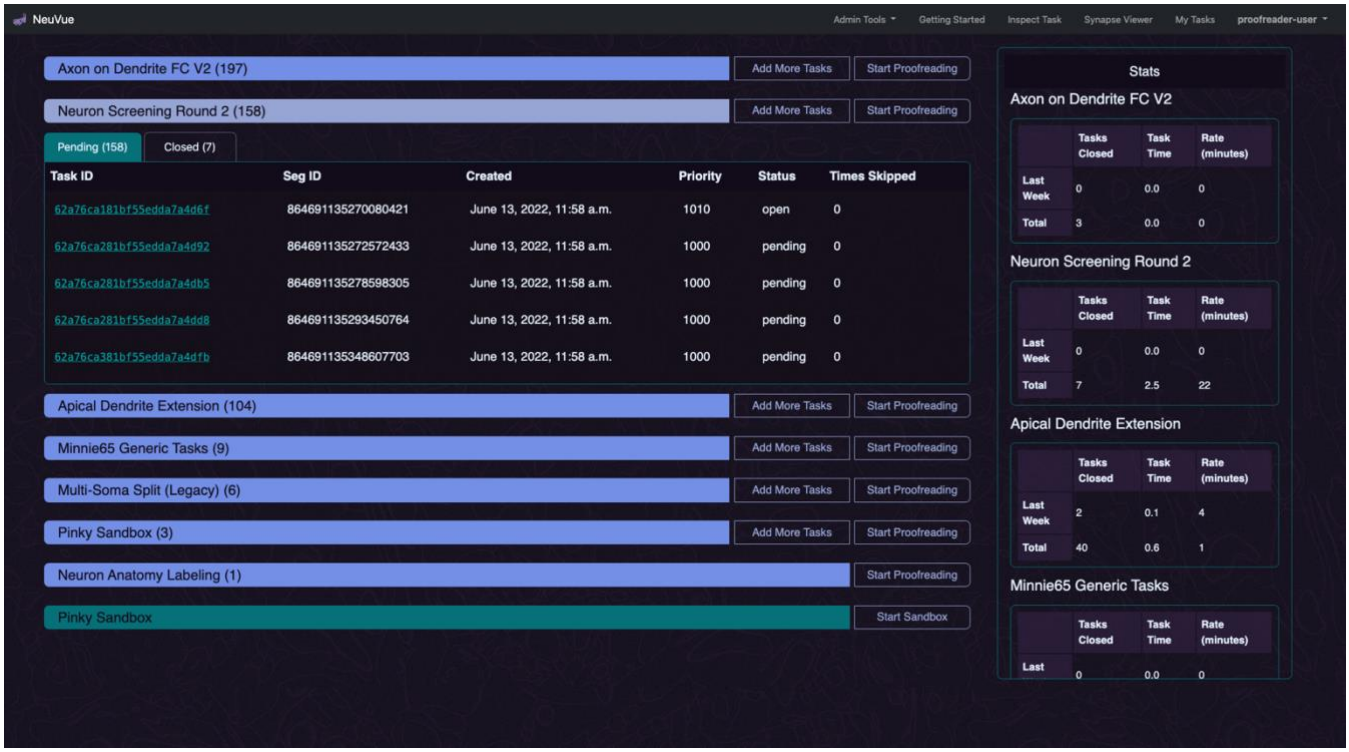

Figure 4. NeuVue task page. This page shows available tasks and user statistics based on past activity. Each blue tab represents an individual task type which can be expanded to show a list of all pending and closed tasks. Buttons located on the right-side of the tab allow users to start proofreading and self-manage their queues.

3.4.2 Workspace Interface and Performing Edits

An authenticated proofreader is routed to the “Workspace” page from the “Task” page provided that they have pending or open tasks to complete. Proofreaders do not get to choose the task they load from the queue; instead, tasks have an administrator-defined priority which determines the order in which they are loaded. The “Workspace” page renders an embedded Neuroglancer window to view and edit the data (Figure 5). The right-side panel displays the relevant task information such as task instructions, relevant segmentation ID, and task tags. Task-specific buttons are located on the top and bottom headers of the screen.

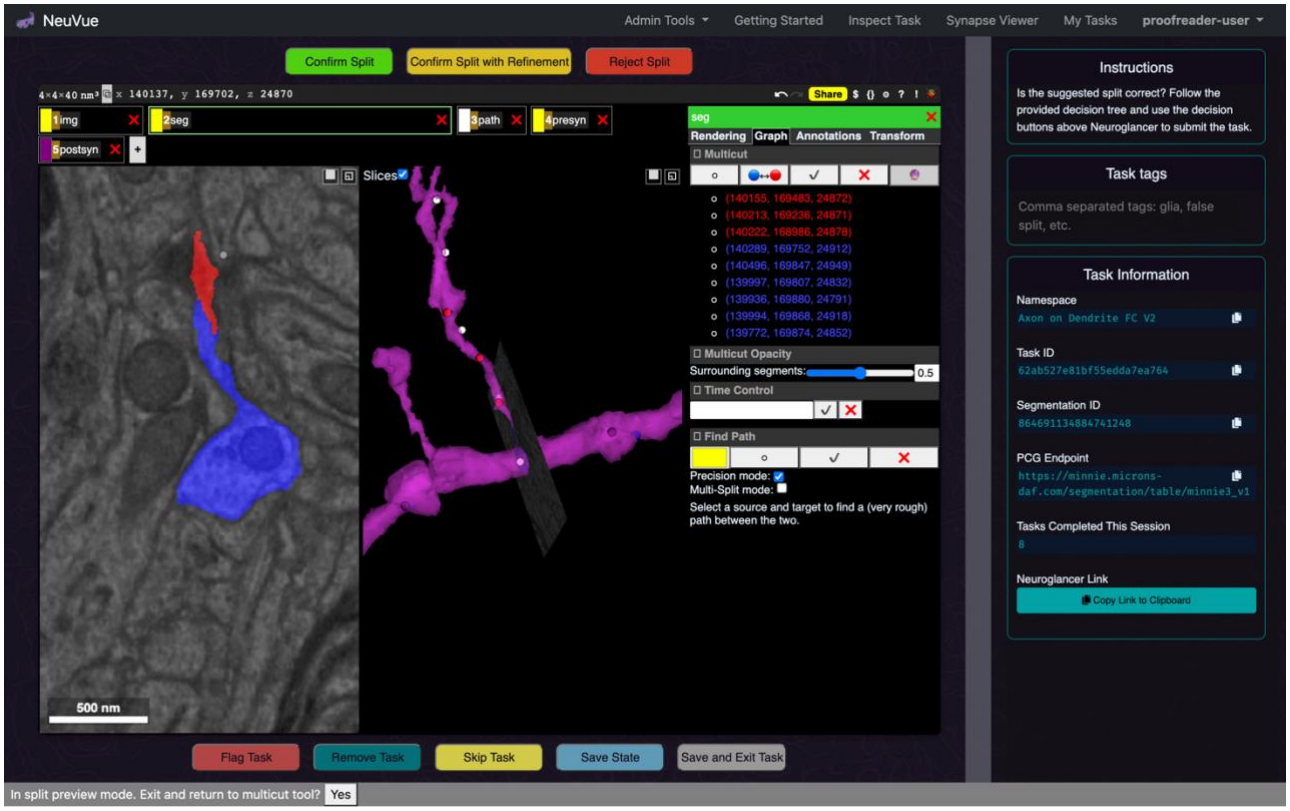

Figure 5. NeuVue workspace page with neuroimaging data loaded (electron microscopy, segmentation), proofreader task actions (buttons at bottom), and task related information (at right). The embedded Neuroglancer window provides the main visualization and editing interface for the proofreader to complete the task. User preferences are available in a submenu which control the layout and UI of the workspace.

Tasks are loaded by retrieving a Neuroglancer state, a JSON object containing data layer and visualization information, from NeuVue-Queue for that task and loading it on the active Neuroglancer window. Once the task loads, the proofreader will follow its instructions and complete the task by one of two methods: by making edits directly in Neuroglancer or by clicking on the administrator-defined task-specific buttons located above the Neuroglancer window.

Proofreaders are able to use Neuroglancer's existing suite of features such as local annotations, hotkeys, and undo/redo to optimize proofreading throughput. We also modified the "NeuVue Neuroglancer", an adapted fork of the Flywire Neuroglancer (Dorkenwald et al. 2022), with additional hotkeys, user preferences, and bug fixes to facilitate a smooth proofreading experience. A link sharing button that copies the state to an external Neuroglancer deployment (https://neuroglancer.neuvue.io) allows tasks to be shared amongst proofreaders and administrators. Besides the context-specific submission methods available to each task types, the proofreader can also perform the following queue-related actions during their task:

- **Flag Task**: Save and close the current task as "errored".
- **Remove Task**: Permanently removes the task from your queue and places it back to the unassigned pool.
- **Skip Task**: Save and re-queue the current task and load next task in the queue.

- **Save State**: Saves the current Neuroglancer state in the Task attributes stored in NeuVue-Queue.
- **Save and Exit Task**: Save the current task and redirect user to the “My Tasks” page. Record total duration.

Proofreaders navigate the data through the Neuroglancer window and, in some cases, make direct changes to the data by using the built-in merge and split tools in the NeuVue Neuroglancer. To keep an accurate log of changes committed by a proofreader, we record the initial and final state of the Neuroglancer viewer, as well as a log of all actions performed by the user. This granularity in user logs creates the ability to quickly debug and resolve issues as well as perform in-depth analysis of proofreader actions.

After a task is submitted, skipped, or flagged, the NeuVue Web App will automatically load the next task for the proofreader if there is one available. During proofreading, NeuVue Web App will record the total active duration, or “session time” a proofreader is on a task. Inactivity reminders and save alerts are also displayed to maintain an accurate record of how much time a proofreader spends on a task.

3.4.3 Administrator Tools

NeuVue contains an admin dashboard, lineage viewer, user management, and task analysis tools for proofreading management. Administrators assign proofreaders into “groups” to facilitate automated task management and separate proofreaders by characteristics such as proofreading proficiency. The admin dashboard provides a tabular summary of proofreader progress for each task type for users in a particular group (Figure 6). Clicking on an individual proofreader yields a secondary table containing information about their assigned tasks. Administrators can directly delete, re-assign, or re-prioritize tasks through this dashboard.

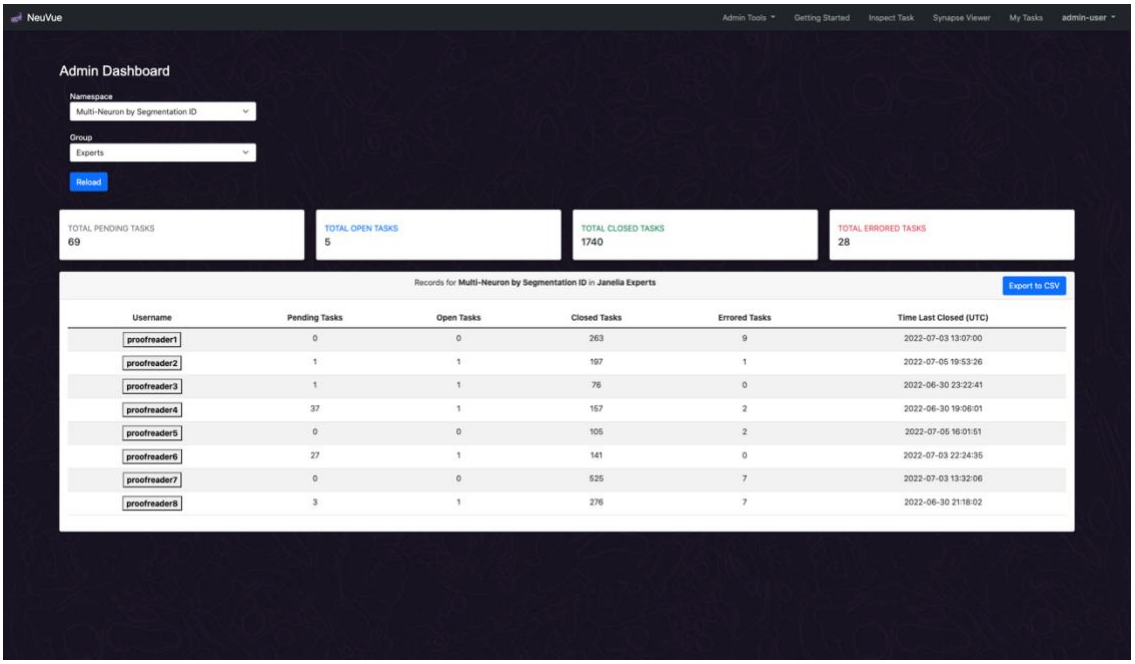

Figure 6. NeuVue admin dashboard allows browsing of proofreader activity, including data on pending, open, closed, and "errored" tasks. Tasks for a specific user open in a separate page if a username is clicked. Dashboards are primarily organized by "Namespace" (task type) and user groups.

Administrators have access to powerful query tools that provide insights on task statistics, which can be filtered by user, time-range, task type, or group through the "Reports" page. For tasks that have a configurable forced-choice decision submission process, bar charts are dynamically generated (Figure 7) showing the distribution of user decisions for a particular set of tasks. These charts and tables can be exported into a sharable image or file.

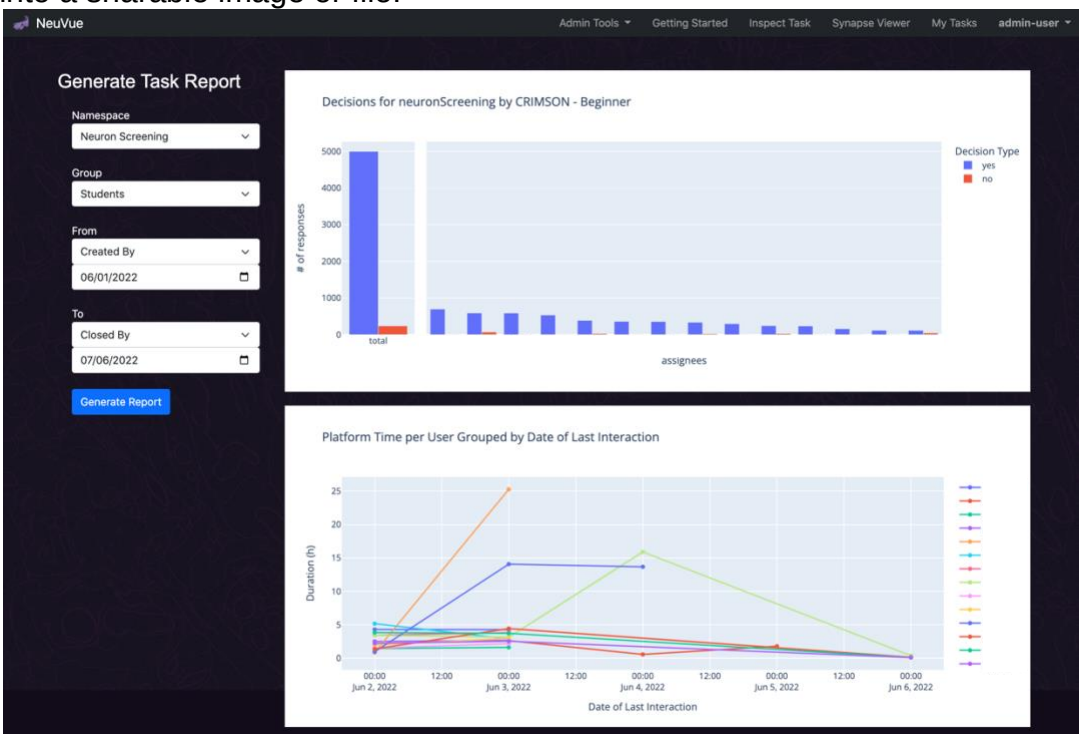

Figure 7. NeuVue reports page allows administrator to generate decision bar charts and visualize proofreading activity over time. Proofreader activity for specific task types and groups can be viewed for user-specific time ranges (user groups/names have been removed for publication).

In order to visualize the proofreading progress in the dataset itself, we developed the "Lineage Viewer" tool (Figure 8). This tool allows administrators to input a neuron's segmentation ID from the dataset and view its current reconstruction as well as previous "history", or associated parent IDs from previous edits, in a single Neuroglancer window. This permits users to intuitively visualize the updates to a particular neuron after proofreading edits have been made.

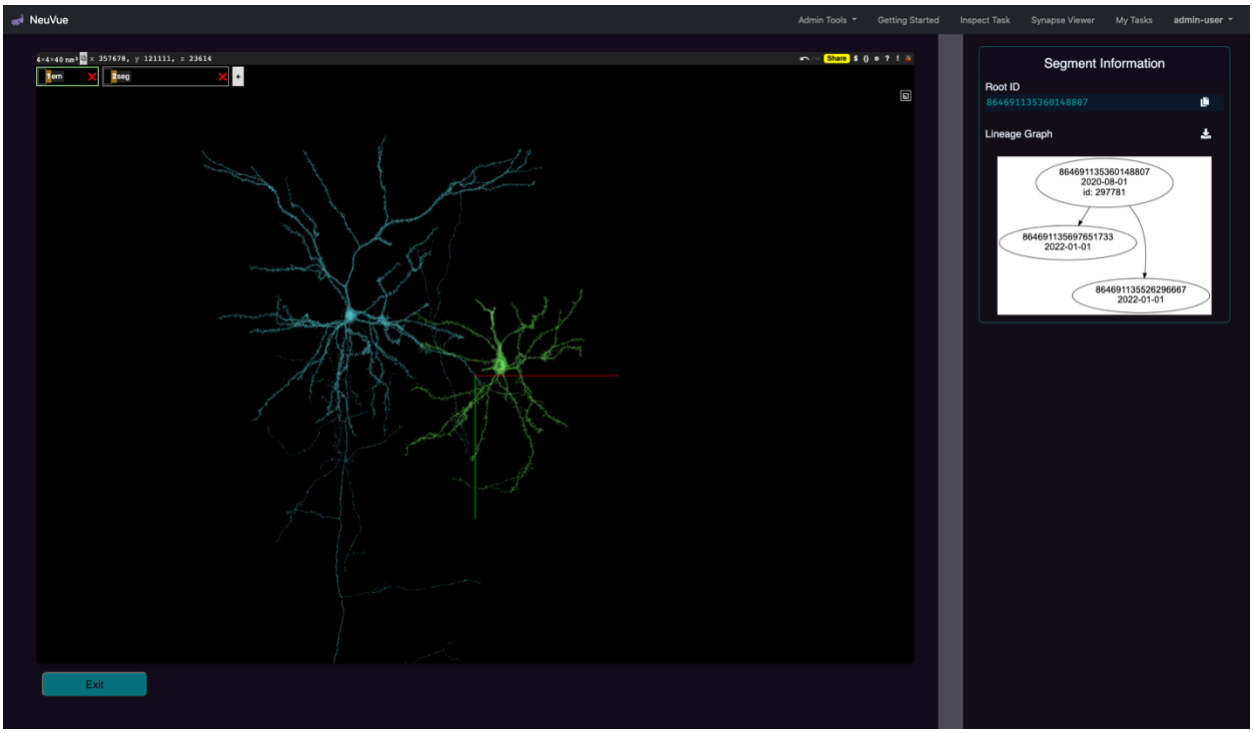

Figure 8. NeuVue lineage viewer permits the visual inspection of neurons that have been proofread, allowing a user to view the state of the neuron at various stages throughout proofreading (with a list of associated IDs at the right). Clicking IDs on the lineage graph image on the right automatically selects that ID in the Neuroglancer window.

4. Results

NeuVue and its integrated queue system were developed to meet the requirements for large-scale proofreading of the MICrONS dataset. These requirements include: 1) compatibility with ongoing proofreading using the PyChunkGraph, 2) high-throughput proofreading, 3) efficient splitting of large merge errors, and 4) ability to leverage innovations from the MICrONS and broader research community.

Our proofreading team consisted of a total of 8 part-time experts with significant previous training and experience in similar tasks; 8 intermediate proofreaders, drawn from a student population who received significant neuroanatomy training and achieved a demonstrated level of proficiency; and 26 novice students .

As an example of NeuVue's capabilities, we describe the proofreading operations optimized for efficient splitting of falsely merged neurons in the MICrONS dataset. Using nucleus detections and neuron classification available for this dataset (MICrONS Consortium et al. 2021), we identified approximately 6,000 segmentation IDs with more than one neuronal nucleus or soma (Figure 9). NeuVue administrators constructed tasks for these segmentation IDs and assigned them to proofreaders.

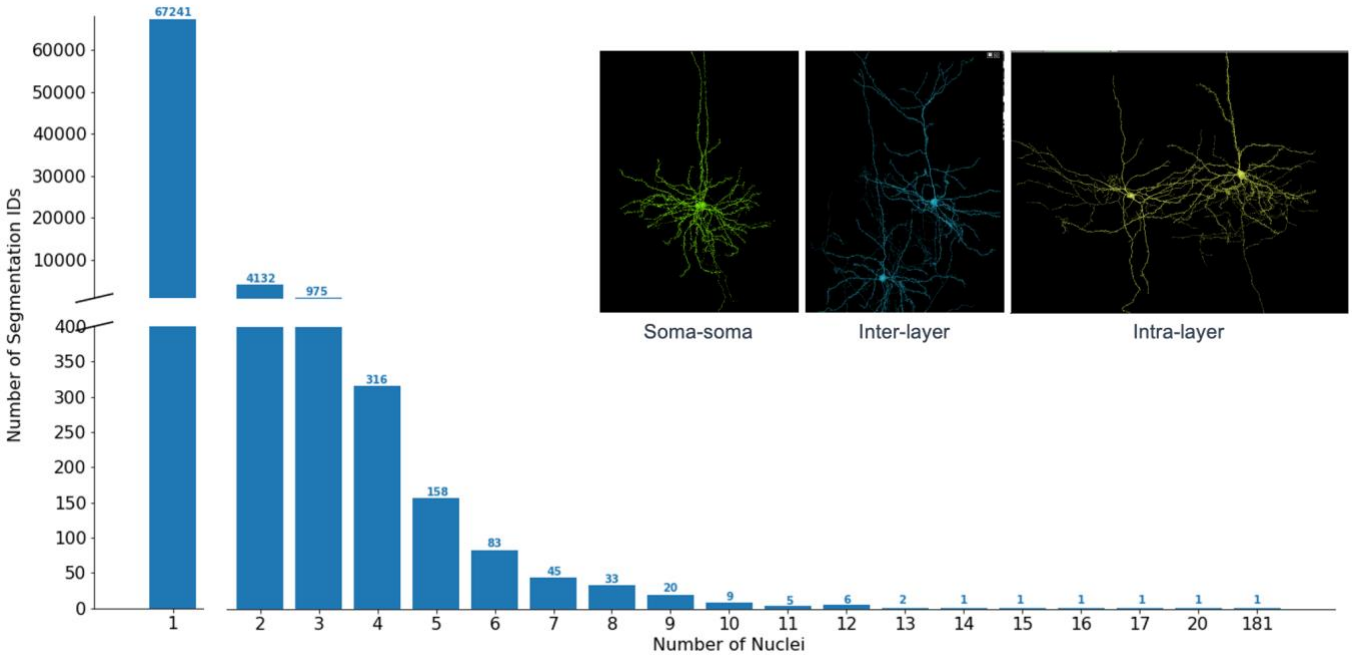

Figure 9. The number of segmentation IDs which contain a certain number of nuclei as of October, 2021. Nuclei detection data were obtained via CAVEclient (Seung-Lab/CAVEclient [2018] 2022) by querying the "nucleus_neuron_svm" table containing predictions of what nuclei in the dataset are neurons.

To date, our proofreaders have successfully split ~5,500 multi-soma segmentation IDs in NeuVue, generating over 14,000 new single-soma neurons. This work was performed in less than one calendar year and required ~3,600 proofreader hours (2.5 person years) to complete. During this process, NeuVue was optimized for both high-throughput validation tasks and semi-automated proofreading tasks as described below.

4.1 Workflow to perform high-throughput validation and correction of automated split locations in multi-soma IDs

To first address multi-soma splitting, we leveraged the results of a newly developed algorithm from Baylor College of Medicine (Celli et al., in preparation) to identify putative merge errors in multi-soma IDs and seed potential split locations. To assess the accuracy of this split location detection algorithm, we examined 9,459 locations that represent putative merge errors in a subset of multi-soma IDs.

For each potential merge error, the split location detection algorithm provided a set of red and blue "split points" which, in combination with a second algorithm, the min-cut algorithm that executes the multi-point cut (Dorkenwald et al. 2022), would split and fix the merge error, and a dotted path indicating the path between the two neurons that would be split (Figure 9). The split location detection algorithm identifies the path between branches of the neurons where there is a merge error. Its accuracy at correctly identifying the location of the merge error on the branch and its accuracy at placing points to split the errors needed to be assessed to determine if it could be run autonomously on a subset of the data volume. To assess the initial accuracy of the algorithm's placement choices, we generated a

"Forced Choice Split Location" proofreading queue to review these locations of interest. To increase the throughput of the assessment, a preview of the proposed split was pre-generated and included in the Neuroglancer state the proofreaders viewed (Figure 10).

4.1.1 Task to identify forced choice split locations

We validated approximately 7,500 putative split locations predicted by the split location detection algorithm using a forced choice-style task. Proofreaders were directed to assess the provided split location and split preview and determine if the location was a true a merge error and if the provided split points correctly resolve the merge error (Figure 10). NeuVue was optimized to provide proofreaders with five responses to choose from:

- **Confirm Split**: if the location and the point placement was adequate
- **Split Requires Refinement**: if the location was correct, but point placement off
- **Reject Split, But Error Nearby**: if the location was incorrect, but a merge error was very nearby
- **Reject Split**: if the location was not a merge error
- **Unsure**: if it was too difficult to assess the location

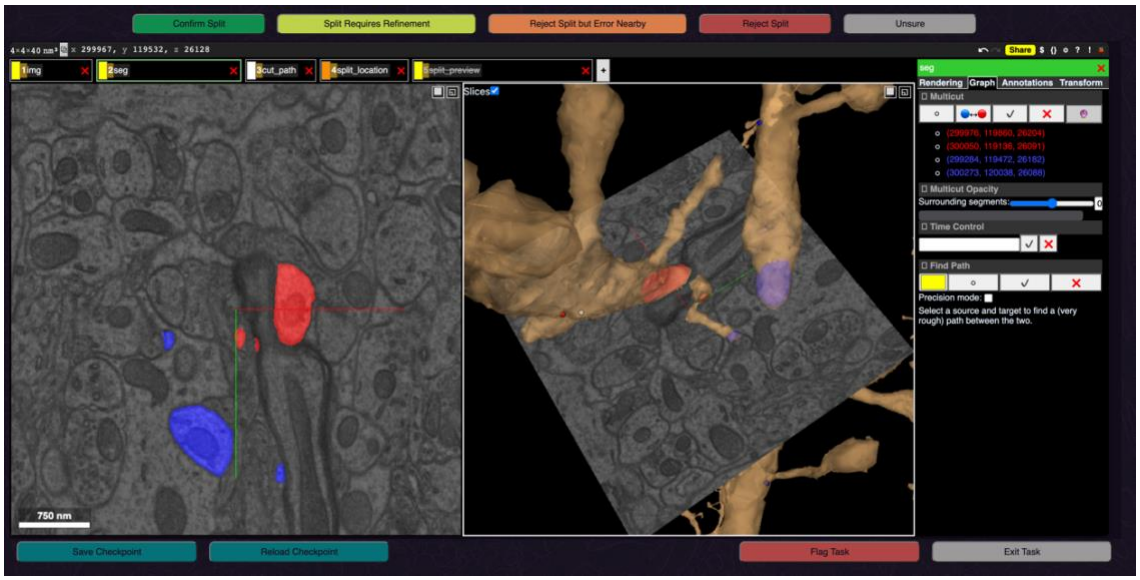

Figure 10. Example of a Forced Choice Task through the NeuVue proofreader web-browser based interface. Forced choice decision buttons are located above the Neuroglancer window. This view currently shows the EM panel with the red and blue colored region indicating how the split would separate the currently selected segmentation ID.

In addition to providing feedback on the algorithm, this exercise helped proofreaders gain familiarity with the data in a low-risk manner. Proofreaders were able to view and assess hundreds of unique error locations in the data before moving on to editing the data.

Depending on the response of the proofreaders to the Forced Choice tasks, the proposed split was then either executed programmatically or placed in one of two additional queues based on difficulty (Figure 11). If the response was 'Confirm Split', the proposed split was considered good as is and the split was executed using the split points provided by the algorithm and the *CAVEclient* API (MICrONS Consortium et al. 2021).

As shown in Figure 11, if the response was 'Split Requires Refinement' or 'Reject Split, Error Nearby' the location was passed to our Task 2 queue: Automated Point Adjustment Queue. If the response was 'Reject Split' or 'Unsure' the location was passed to our Task 3 queue: Guided Find and Fix.

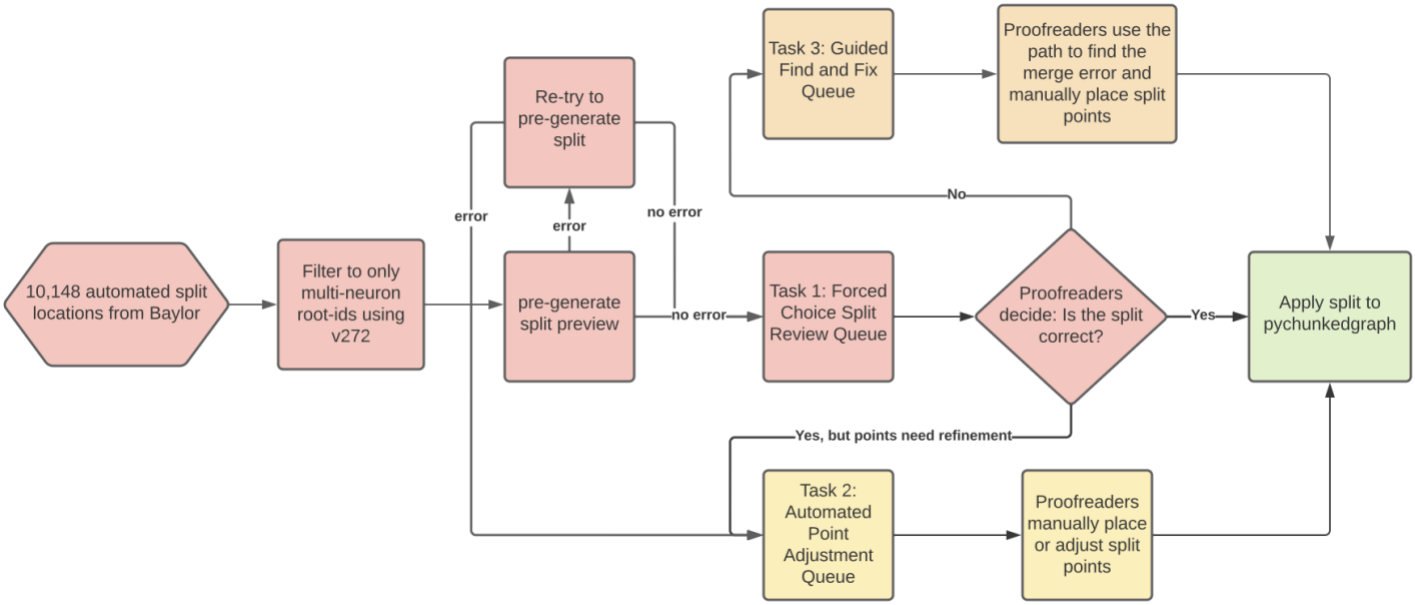

Figure 11. Proofreading workflow for the automated split locations from Baylor College of Medicine Automated Proofreading Algorithm. After filtering split locations provided by Baylor College of Medicine algorithm and pre-generating the split preview, 7,515 split locations were available for task 1.

4.1.1.1 Force Choice Proofreader Throughput

To measure forced choice proofreader throughput, the average time per task was computed for each task in this queue (Figure 10). Based on a subset of 4,181 tasks completed by experienced proofreaders, the average time per task was 1.52 minutes (Figure 12). Interestingly, the average time per task for novice proofreaders was also 1.56 minutes (Figure 12), which was similar in time to the expert proofreaders.

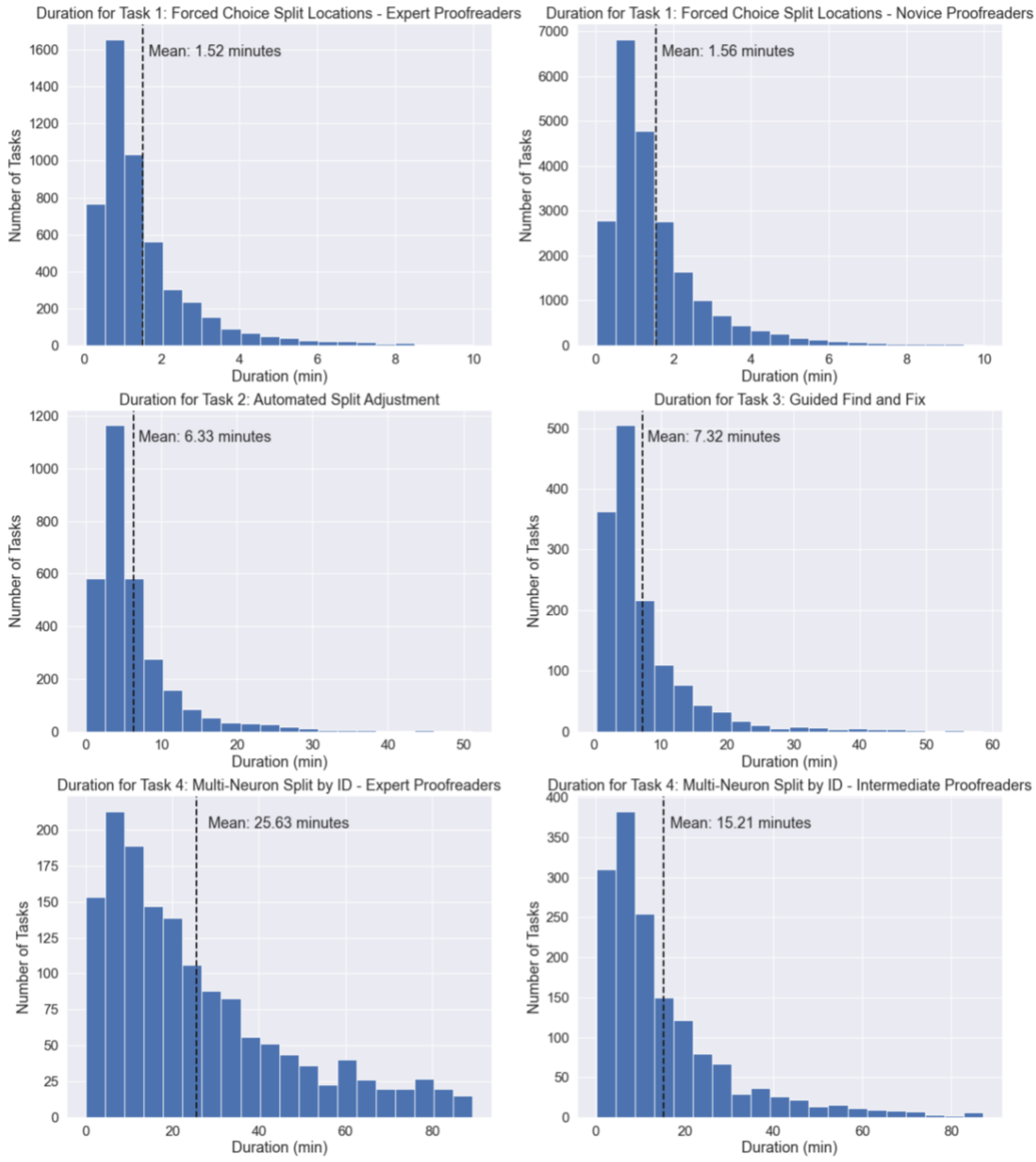

Figure 12. Duration of Proofreading Tasks. Histograms of time to complete task in minutes. Top row: Forced Choice Split Location tasks by expert (left) and novice (right) proofreaders; Middle row: Automated Point Adjustment task (left) and Guided Find and Fix task (right); both by expert proofreaders only; Bottom row: Multi-Neuron by Segmentation ID by expert (left) and intermediate (right) proofreaders.

4.1.2 Task to adjust automated split points

Our Automated Point Adjustment queue addresses the locations from the split detection algorithm that either correctly identified the merge error location but misidentified the split points that would fix the

merge, or were very near to a merge error. Proofreaders were provided the Neuroglancer view of the split location and the automatically placed points and were instructed to manually adjust or replace the red and blue split points to correctly split the merge error (as judged by previewing the split using the min-cut tool in Neuroglancer). They then executed the split directly in Neuroglancer. Once the merge error was split, they submitted the task and were provided the next split location.

To measure proofreader throughput, we computed the average time per task for this queue (Figure 10). Given that this operation requires manually adjusting of the split points and may require multiple attempts, the average time per task was 6.8 minutes (based on 1,662 tasks executed by expert proofreaders).

4.1.3 Task to find and fix locations with guidance

For our Guided Find and Fix queue, proofreaders review the locations where the algorithm did not find the correct merge location or it was too difficult or complicated to assess quickly. These tasks are more difficult than the Task 2 locations due to the nature of the merge errors that were missed by the algorithm. Frequently, the merge errors that end up in this queue are locations where two different branches were incorrectly merged 'head-to-head' due to the shifts in image alignment (Figure 13). These are relatively difficult to identify from a distance and require careful stepping through the provided cut path. Proofreaders were instructed to follow the provided cut path, find the merge error, place the necessary red/blue split points, preview the split, and manually execute the split in Neuroglancer.

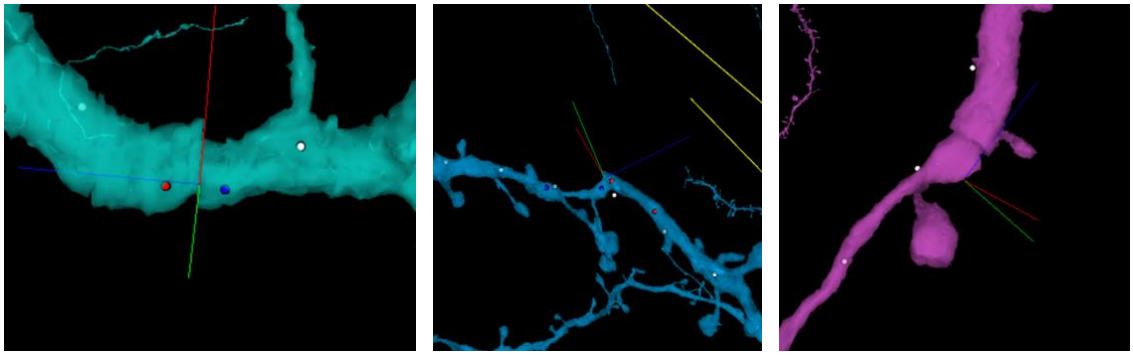

Figure 13. Examples of 'head-to-head' merge errors in the Task 3 queue.

We computed the average time per task for this queue based on 884 closed tasks. Similar to the Points Adjustment task, this task requires manually adjusting the split points and may require multiple attempts. This work was performed by experienced proofreaders. The average time per task was 6.5 minutes (Figure 12).

4.2 Workflow to perform semi-automated splitting of multi-soma IDs

4.2.1 Task to split multi-neurons by segmentation ID

Multi-soma IDs often contain several merge errors (up to 20 per ID) that must be corrected to successfully isolate single-soma neurons. Using the Split Detection Algorithm described in section 4.1, we generated approximately 2,500 new single-soma neurons and reduced the number of multi-soma IDs by about 1,000. For the remaining 5,000 multi-soma IDs in the dataset, the algorithm predicted split locations were not available. To accommodate this, a targeted and semi-automated approach to "comprehensively" correct merge errors was applied for each multi-soma ID until single-soma IDs were generated. Proofreaders generated the dotted path on demand using the path tool in Neuroglancer (Dorkenwald et al. 2022) and inspected the path for merge errors (Figure 14). During the forced choice tasks, proofreaders evaluated hundreds of unique error locations in the data and thus became familiar with how to visually identify merge errors.

After locating the merge error, proofreaders manually placed split points and previewed the split using Neuroglancer's min-cut tool (Dorkenwald et al. 2022), as done in the Automated Point Adjustment task. If the preview was acceptable, the proofreader applied the split operation to the dataset through Neuroglancer. Proofreaders also had the option to adjust the split points as needed to achieve an optimal split. Neuroglancer refreshed after each split operation and two single-soma IDs were generated if there were no remaining merge errors between soma. If not, proofreaders generated another dotted path and repeated the process until two IDs were generated.

To train and evaluate our novice proofreaders, we prepared a ground truth set for forced choice and split point adjustment tasks. Proofreaders with ≥ 90% agreement with ground truth were approved to work on multi-soma splitting, and received edit permission. We also recorded the operation ID of proofreader edits made on the segmentation, so that administrators could review and reverse edits if necessary.

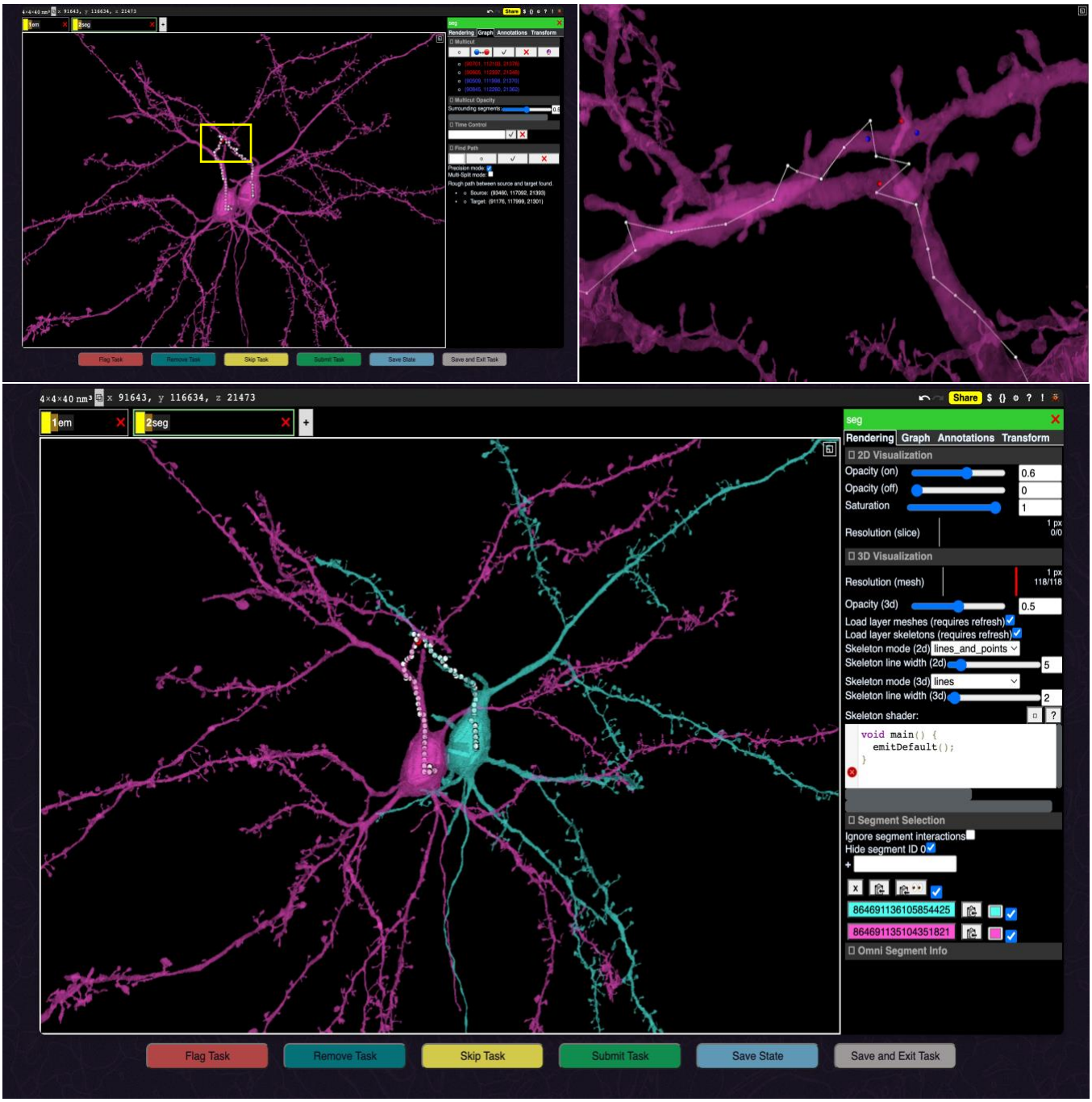

Figure 14. Multi-Neuron by Segmentation ID task. Top row: dotted path & split location with red and blue points; Bottom row: neurons appear as different colors with new IDs if successfully separated by split operation.

4.3 Multi-Soma Splitting Throughput

To date, we have split 5,460 multi-soma IDs with more than one neuronal nucleus and generated 14,426 new single-soma neurons. This work was completed by 8 part-time experts with proofreading

experience in Drosophila connectomics, and 8 intermediate-level proofreaders. Given their intermediate level of experience, non-experts primarily worked on multi-soma with no more than two nuclei, and the most difficult multi-soma IDs were reserved for experts.

Using NeuVue's dashboard, we computed the total proofreading time to complete all tasks in each queue (Table 1). In total, proofreaders completed ~36,000 tasks and this work required ~3,600 hours. It took four times as much proofreading time per single-soma neuron using the split detection algorithm (~40 minutes) versus semi-automated split detection (~10 minutes; Table 2). This difference is largely due to 1) assigning same split location to multiple novice proofreaders for consensus decision-making, and 2) improving efficiency for expert- and intermediate-level proofreading by consolidating operations (split location and point adjustment) into one task and making updates to NeuVue. It is important to note that in our initial use case, we were intentionally conservative with novice proofreaders and assigned 5-6 proofreaders to each split location as we assessed their ability to work on more complicated tasks.

We also measured proofreader throughput by computing average time per task for this queue. Based on 1,628 tasks completed by expert proofreaders, the average time per task was 25.6 minutes (Figure 12). As expected, the average time per task for intermediate proofreaders was less (15.6 minutes; based on 1,606 tasks) because they worked exclusively on multi-soma IDs with two nuclei while experts worked on multi-soma IDs with up to 20 nuclei (Figure 12).

Split Location Detection Algorithm	Task	Total Duration (hours)	Proofreader Level	# Split Locations	# Tasks Completed
Yes	Forced Choice Split Locations	208.37	Expert	5,139	5,139
Yes	Forced Choice Split Locations – Group[1]	827.5	Novice	3,086	22,967
Yes	Automated Split Points Adjustment	367.31	Expert	3,059	3,059
Yes	Guided Find and Fix the Merge Error	184.09	Expert	1,421	1,421
No	Multi-Neuron by Segmentation ID[2]	1,441.4	Expert	n/a	1,628
No	Multi-Neuron by Segmentation ID[2]	567.49	Intermediate	n/a	1,606

[1]Decision by consensus (26 novice proofreaders); 5-6 proofreaders per split location; one split location per task

[2]Proofreaders corrected one or more split locations per segmentation id; one id per task

Table 1. Total time to complete all tasks per queue (with or without split location detection algorithm).

Split Location Detection Algorithm	Total Duration (hours)	# Single-soma Neurons Generated	Time per Single-soma Neuron (hours)
Yes	1,587.3	2,468	0.64
No	2,008.89	12,152	0.17

Table 2. Total time to complete all tasks and time per single-soma neuron generated (with and without split location detection algorithm).

5. Discussion

We developed a lightweight, web-based framework to guide proofreaders from heterogeneous expertise levels through high-value workflows to mature the quality of the connectomics data produced through machine learning methods. These tools and accompanying workflows leverage efficient task queuing and customized, atomic environments to aid users in making rapid, atomic, and high-quality decisions. When considering the rapid advances in technology and algorithms that enable large-scale image acquisition and automated alignment, registration, and segmentation approaches, proofreading strategically and scalably remain as major contemporary bottlenecks. Reducing the human resource expense (e.g., time, cost, available workforce) is critical to many future research questions. We believe that these tools provide an important, complementary toolkit to existing methods and can be easily expanded to incorporate additional datasets, tasks, and user communities.

NeuVue was developed to meet the demands for large-scale proofreading of the MICrONS dataset, one of the largest connectomic datasets produced to date with automatic machine segmentation (MICrONS Consortium et al. 2021). In general, segmentation errors can be divided into two types: merge errors, in which two or more neurons are falsely merged together, and split errors, in which a neuron is falsely split or broken, resulting in orphan segments. Machine segmentation typically begins with an over-segmented base segmentation followed by agglomeration of segments into neurons. A conservative agglomeration may minimize merge errors at the expense of generating more orphans, such as broken axons. In contrast, an aggressive agglomeration that minimizes orphans can be applied if resulting merge errors can be efficiently corrected, as done in the hemibrain dataset (Scheffer et al. 2020). The hemibrain volume is approximately 1/64th of the MICrONS volume and another recently released cubic millimeter volume, H01 (Shapson-Coe et al. 2021). Correcting merge errors in increasingly larger volumes presents a major proofreading challenge. H01 used a conservative agglomeration that minimized merges between neuronal somata at the expense of more axon breaks (Shapson-Coe et al. 2021). NeuVue's tools and multi-soma splitting workflows may allow researchers to use a more aggressive agglomeration in the next generation of MICrONS- and H01-like or larger volumes, thus allowing more axonal connections between neurons to be analyzed by machine segmentation.

6. References

Bishop, Caitlyn, Jordan Matelsky, Miller Wilt, Joseph Downs, Patricia Rivlin, Stephen Plaza, Brock Wester, and William Gray-Roncal. 2021. “CONFIRMS: A Toolkit for Scalable, Black Box Connectome Assessment and Investigation.” In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2444–50. https://doi.org/10.1109/EMBC46164.2021.9630109.

Boergens, Kevin M., Manuel Berning, Tom Bocklisch, Dominic Bräunlein, Florian Drawitsch, Johannes Frohnhofen, Tom Herold, et al. 2017. “WebKnossos: Efficient Online 3D Data Annotation for Connectomics.” *Nature Methods* 14 (7): 691–94. https://doi.org/10.1038/nmeth.4331.

Dorkenwald, Sven, Claire E. McKellar, Thomas Macrina, Nico Kemnitz, Kisuk Lee, Ran Lu, Jingpeng Wu, et al. 2022. “FlyWire: Online Community for Whole-Brain Connectomics.” *Nature Methods* 19 (1): 119–28. https://doi.org/10.1038/s41592-021-01330-0.

Dorkenwald, Sven, Nicholas L. Turner, Thomas Macrina, Kisuk Lee, Ran Lu, Jingpeng Wu, Agnes L. Bodor, et al. 2019. “Binary and Analog Variation of Synapses between Cortical Pyramidal Neurons.” *BioRxiv*, December, 2019.12.29.890319. https://doi.org/10.1101/2019.12.29.890319.

Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

Hider, R. *et al.* The Brain Observatory Storage Service and Database (BossDB): A Cloud-Native Approach for Petascale Neuroscience Discovery. *Front Neuroinform* 16, (2022).

Hubbard, Philip M., Stuart Berg, Ting Zhao, Donald J. Olbris, Lowell Umayam, Jeremy Maitin-Shepard, Michal Januszewski, William T. Katz, Erika R. Neace, and Stephen M. Plaza. 2020. “Accelerated EM Connectome Reconstruction Using 3D Visualization and Segmentation Graphs.” Preprint. Neuroscience. https://doi.org/10.1101/2020.01.17.909572.

Januszewski, Michał, Jörgen Kornfeld, Peter H. Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. 2018. “High-Precision Automated Reconstruction of Neurons with Flood-Filling Networks.” *Nature Methods* 15 (8): 605–10. https://doi.org/10.1038/s41592-018-0049-4.

Kim *et al.* Space–time wiring specificity supports direction selectivity in the retina. *Nature* **509**, 331–336 (2014).

Maitin-Shepard, Jeremy. Neuroglancer. https://github.com/google/neuroglancer. Retrieved 2017-06-10.

MICrONS: Machine intelligence from cortical networks. http://iarpa.gov/index.php/research-programs/microns. Retrieved 2017-10-31.

MICrONS Consortium, J. Alexander Bae, Mahaly Baptiste, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, et al. 2021. “Functional Connectomics Spanning Multiple Areas of Mouse Visual Cortex.” Preprint. Neuroscience. https://doi.org/10.1101/2021.07.28.454025.

Plaza, S. M. Focused Proofreading: Efficiently Extracting Connectomes from Segmented EM Images. (2014) doi:10.48550/ARXIV.1409.1199.

Popovych, Sergiy, Thomas Macrina, Nico Kemnitz, Manuel Castro, Barak Nehoran, Zhen Jia, J. Alexander Bae, et al. 2022. “Petascale Pipeline for Precise Alignment of Images from Serial Section Electron Microscopy.” *BioRxiv*, March, 2022.03.25.485816. https://doi.org/10.1101/2022.03.25.485816.

Reback, J. *et al. pandas-dev/pandas: Pandas 1.4.3*. (Zenodo, 2022). doi:10.5281/ZENODO.6702671.

Saalfeld, Stephan, Albert Cardona, Volker Hartenstein, and Pavel Tomančák. 2009. “CATMAID: Collaborative Annotation Toolkit for Massive Amounts of Image Data.” *Bioinformatics* 25 (15): 1984–86. https://doi.org/10.1093/bioinformatics/btp266.

Scheffer, Louis K, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, et al. 2020. “A Connectome and Analysis of the Adult Drosophila Central Brain.” Edited by Eve Marder, Michael B Eisen, Jason Pipkin, and Chris Q Doe. *ELife* 9 (September): e57443. https://doi.org/10.7554/eLife.57443.

Seung-Lab/CAVEclient. (2018) 2022. Python. seung-lab. https://github.com/seung-lab/CAVEclient.

Shapson-Coe, Alexander, Michał Januszewski, Daniel R. Berger, Art Pope, Yuelong Wu, Tim Blakely, Richard L. Schalek, et al. 2021. “A Connectomic Study of a Petascale Fragment of Human Cerebral Cortex.” Preprint. Neuroscience. https://doi.org/10.1101/2021.05.29.446289.

Turner, Nicholas L., Thomas Macrina, J. Alexander Bae, Runzhe Yang, Alyssa M. Wilson, Casey Schneider-Mizell, Kisuk Lee, et al. 2022. “Reconstruction of Neocortex: Organelles, Compartments, Cells, Circuits, and Activity.” *Cell* 185 (6): 1082-1100.e24. https://doi.org/10.1016/j.cell.2022.01.023.

Yin, Wenjing, Derrick Brittain, Jay Borseth, Marie E. Scott, Derric Williams, Jedediah Perkins, Christopher S. Own, et al. 2020. “A Petascale Automated Imaging Pipeline for Mapping Neuronal Circuits with High-Throughput Transmission Electron Microscopy.” *Nature Communications* 11 (1): 4949. https://doi.org/10.1038/s41467-020-18659-3.

Zhao, Ting, Donald J. Olbris, Yang Yu, and Stephen M. Plaza. 2018. “NeuTu: Software for Collaborative, Large-Scale, Segmentation-Based Connectome Reconstruction.” *Frontiers in Neural Circuits* 12 (November). https://doi.org/10.3389/fncir.2018.00101.

7. Acknowledgements

The authors thank David Markowitz, the IARPA MICrONS Program Manager, who coordinated the activities and developments presented in this paper. We thank the MICrONS Proofreading Coordination Group (Forrest Collman, Casey Schneider-Mizell, Nuno da Costa, Jacob Reimer, Andreas Tolias, Brendan Celli, Paul Fahey, Sven Dorkenwald) for sharing tools and providing feedback, and, in particular, Brendan Celli and Jacob Reimer for providing results of their split location detection algorithm for testing. We also thank A. Mendpara, C. Oji, C. Zhang, C. Moore, D. Parodi, E. Yan, J. Chan, J. Simmons, K. Zhang, K. Patel, L. Zhang, M. Mitchell, M. Amosu, R. Xu, R. Chalavadi, S. Hong, T. Gaito, T. Xie, T. Carruth, V. Lung, Z. Ekinci, S. Bare, S. Wu, S. Darcy, L. Fozo, C. Ordish, C. Knecht, E. Neace, M. Cook, G. Hopkins, E. Phillips, N. Smith, C. Smith for proofreading, and Jordan Matelsky and Caitlyn Bishop for providing computational support as well as sharing their insights from previous phases of the MICrONS project.