BMC Bioinformatics

RESEARCH

Open Access

Toward a more accurate 3D atlas of C. *elegans* neurons

Michael Skuhersky[1*], Tailin Wu[2], Eviatar Yemini[3], Amin Nejatbakhsh[4,5], Edward Boyden[1,6] and Max Tegmark[7]

*Correspondence:
vex@mit.edu

[1] Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, USA
Full list of author information is available at the end of the article

Abstract

Background: Determining cell identity in volumetric images of tagged neuronal nuclei is an ongoing challenge in contemporary neuroscience. Frequently, cell identity is determined by aligning and matching tags to an "atlas" of labeled neuronal positions and other identifying characteristics. Previous analyses of such *C. elegans* datasets have been hampered by the limited accuracy of such atlases, especially for neurons present in the ventral nerve cord, and also by time-consuming manual elements of the alignment process.

Results: We present a novel automated alignment method for sparse and incomplete point clouds of the sort resulting from typical *C. elegans* fluorescence microscopy datasets. This method involves a tunable learning parameter and a kernel that enforces biologically realistic deformation. We also present a pipeline for creating alignment atlases from datasets of the recently developed NeuroPAL transgene. In combination, these advances allow us to label neurons in volumetric images with confidence much higher than previous methods.

Conclusions: We release, to the best of our knowledge, the most complete full-body *C. elegans* 3D positional neuron atlas, incorporating positional variability derived from at least 7 animals per neuron, for the purposes of cell-type identity prediction for myriad applications (e.g., imaging neuronal activity, gene expression, and cell-fate).

Keywords: *Caenorhabditis elegans*, Neuron identification, Point-cloud alignment, Cell atlas

Background

The nematode Caenorhabditis elegans (*C. elegans*) is among the most-studied animals in neuroscience, and remains the only multicellular organism with a fully mapped connectome. Capable of exhibiting complex behaviors despite having only several hundred neurons, it has provided an abundance of neuroscientific insights. The goal of this paper is to further improve the scientific utility of *C. elegans* by enabling more accurate and automated identification of its neurons, that are either unlabeled, or have been labeled by one or more colors that encode information regarding identity.

Limitations of the original *C. elegans* connectome

The electron micrograph (EM) reconstruction of the *C. elegans* nervous system and its connectome were first fully described in a seminal 1986 paper [1]. This was an invaluable technical achievement requiring a decade of work to hand-trace every neuron and connection from the EM sections. Due to technical limitations involved with preparing worm samples, this nervous system reconstruction was derived from a mosaic of overlapping sections from five individual worms. These five worms consist of three adult hermaphrodites, a fourth larval stage (L4) animal, and one adult male. Thus, in combination, this reconstruction provides a generalized view of the worm's nervous system.

While a generalized view of the worm's nervous system has proven valuable to the field, it lacks representation for the idiosyncrasies found among individual worms. Moreover, preparations for EM imaging can introduce non-linear distortions. Preparing worms for EM imaging requires that they be first physically sliced open so that the fixative may bypass the impermeable cuticle [2, 3]. As the worm interior is under a higher pressure, breaching the cuticle results in morphological changes, and the commonly used EM fixative osmium tetroxide has also been shown to alter morphology [4]. In the original reconstruction of the worm nervous system, animals were serially sectioned into approximately 50 nm thick slices. The combination of these steps yielded distortions that required correction when unifying worm sections into a general visual representation of its nervous system. As a result, substantial manual correction was introduced when generating this canonical nervous system illustration. These corrections and the multi-animal synthesis thus present a generalized view of the worm's nervous system, albeit one that lacks quantification of neuron positions and measurements of their variability to recapitulate individual idiosyncrasies present within the population.

Existing *C. elegans* atlases

This illustrative 1986 worm atlas has often been treated as canon, or at least as an atlas of sufficient quality to compare with contemporary data from modern and more reproducible measurement techniques. This is in large part due to it being the only atlas of its kind up until very recently [5, 6]. For example, Scholz et al. [7] aligns this atlas to fluorescent imaging data to assign neuron identity. Unfortunately, using this generalized illustrative atlas for neural identification purposes can lead to unlabeled and even mislabeled neurons. This is largely due to the density of neurons in various ganglia and ambiguities in atlas matching that are present as a result. In addition to these limitations and those listed in the previous section, a 2004 version of this atlas, commonly used in papers that make use of *C. elegans* neuron positions, was produced by further processing the original version so as to translate the 1986 illustration into semi-quantifiable measurements. This version was produced by Choe et al. [8, 9] and popularized by Kaiser et al. [10]. It tabulates neural positions that were measured by directly scanning and tracing the physical 1986 paper. The 2004 atlas (Fig. 1) consists of only 277 of the 302 neurons found in adult hermaphrodites. Crucially, due to the 2D geometry of the scanned and processed original paper, the 2004 atlas is also 2D.

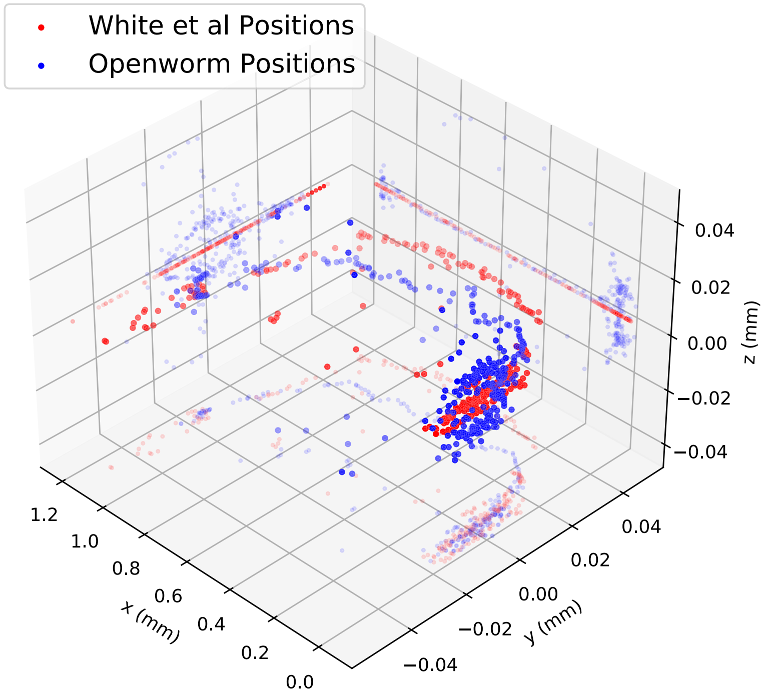

Fig. 1 *C. elegans* atlases from White et al. (red) and OpenWorm (blue), after straightening and uniform scaling using a corrected version of [21], with axial projections, unequal scaling

The third dimension in modern 3D datasets is therefore typically discarded before matching against this 2D atlas and may further increase the misidentification rate. In total, these limitations motivate the creation of a representative 3D atlas of *C. elegans* neuron positions and their variability.

In the decades since the original EM connectome was published [1], many individual *C. elegans* neurons have been further studied and characterized. Their updated information has been incorporated into the widely used *C. elegans Atlas* [11], and the contents of this reference text have been artistically assembled into the popular and influential OpenWorm project [12, 13]. OpenWorm presents information for individual neurons and their connectivity, assembled into a 3D approximation of an adult hermaphrodite worm. However, the inherent positional variability of neurons in the head and tail is now readily apparent from multicolor *C. elegans* strains, designed for neural identification, that were used to measure neuron positions and their variability across multiple animals [5, 6, 14]. These strains and concurrent algorithmic advances demonstrate how fluorescent-protein barcodes can be used to accurately determine neuron identities in volumetric images [15–20].

The strong interest in *C. elegans* research coupled with great recent progress in all aspects of *C. elegans* imaging makes it timely to develop improved tools for whole-nervous-system *C. elegans* neuron identification and make them available to the community. This is the goal of the present paper. The rest of this paper is organized as follows: we present our improved alignment method in "Methods" together with a pipeline for creating improved atlases of whole-worm neuron positions. We then compare the neuron identification accuracies of various methods in "Results",

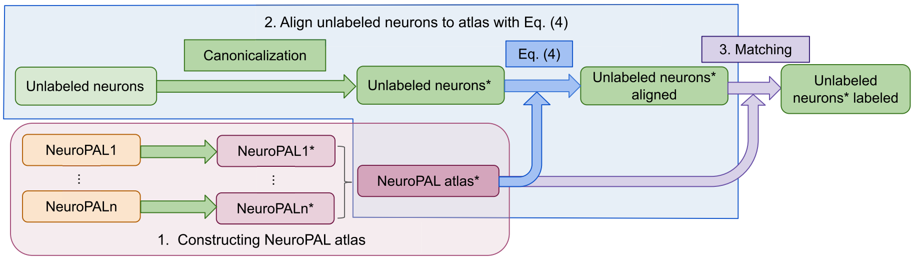
2. Align unlabeled neurons to atlas with Eq. (4)
Canonicalization
Unlabeled neurons
Unlabeled neurons*
Eq. (4)
Unlabeled neurons* aligned
3. Matching
Unlabeled neurons* labeled
NeuroPAL1
NeuroPAL1*
NeuroPALn
NeuroPALn*
NeuroPAL atlas*
1. Constructing NeuroPAL atlas

Fig. 2 Diagram of our NeuroPAL atlas construction and alignment pipeline, with each algorithm labeled. The * denotes the neuron positions after canonicalization

applying them to simulated data and then to our new atlas. We then summarize these results in our "Conclusion".

Methods

New NeuroPAL-derived atlas

Our construction of the NeuroPAL atlas consists of three steps:

1. Preparing and imaging NeuroPAL worms and turning each image into a pair of measured quantities as in "Preparing and imaging the NeuroPAL worms": a point cloud (a position vector and optional color information for each neuron) and a hull outline (a smooth closed 2D curve).
2. Canonicalizing each pair by normalizing the hull into a standard straightened hull, along with the point cloud in the associated canonical 3D space.
3. Combining point clouds from multiple worms into a single point cloud atlas (Fig. 2).

Preparing and imaging the NeuroPAL worms

Brainbow [22] is a stochastic technique that has been used to differentiate individual neurons from neighboring ones by expressing unique ratios of red, green, and blue fluorescent proteins. Unfortunately, Brainbow coloring is generated randomly and thus the colors cannot be used to identify neuron types. In contrast, the recent NeuroPAL *C. elegans* transgene introduces an alternative deterministic technique that reveals the unique identity of each individual neuron, at all larval stages of both worm sexes (hermaphrodites and males) [6, 23]. NeuroPAL worms have an identical and invariant colormap by way of the stereotyped expression of four distinguishable fluorophores. These four fluorophores leave the green channel free, and thus it can be used to map gene expression using the common GFP, CFP, or YFP based gene reporters, or to measure dynamic neural activity using the GCaMP reporter.

A complicating factor in position-derived identification is that individual neuron positions are known to be locally variable between different worms [5, 6, 11]. This is caused partly by *C. elegans* size and shape differences, and partly because of inherent positional variance that may be so extreme as to have nearby neurons switch relative positions.

Adult *C. elegans* were imaged in accordance with the methods as described in [6], and the images were processed with a semiautomatic method using accompanying software of [6]. Neuron positions were detected by the software, and any required corrections were made manually. The software predicted neuron identities for the head and tail of each worm, and these were manually checked and relabeled as needed. Since there were no prior statistical atlases for neurons in the worm midbody, the identities of these neurons were manually annotated for each worm. The result was a labeled point cloud for each individual corresponding to the neuron positions.

The original 1986 nervous system reconstruction [1] was assembled using a mosaic of 5 overlapping image sections, each corresponding to part of the worm, that when pieced together would represent entire nervous system. Each section was taken from one of 5 worms representing a mixture of age and sex: 3 adult hermaphrodites, 1 L4, and 1 adult male. In our atlas, to maintain a generalized representation of the nervous system we used 7 worms: 1 adult hermaphrodite, 4 young-adult hermaphrodites, 1 L4 hermaphrodite, and 1 adult male. Here, each individual worm has neuron positional information represented for its whole body, rather than only a body section, thus contributing a holistic representation of the nervous system from each of these animals. All 7 worms were positioned on an agar pad for imaging such that their left-right axis extended between the glass coverslip and slide. As a result, the worm samples and representative left-right axis may have been slightly compressed between these two surfaces. Similarly, the dorsal-ventral and potentially the anterior-posterior axes may have been slightly elongated as a result of this compression.

Straightening method

For successful alignment, canonicalization is essential in ensuring that the neuron point clouds from different worms, that were imaged in different postures, orientations, and morphology, lie in the same canonical space to provide a reasonable starting point for neuron matching. Mathematically, canonicalization corresponds to a continuous and invertible 3D mapping that that gives all imaged worm hulls the same shape regardless of the proportions and bending state of the worms.

Several methods have been previously used for this problem of straightening worm hulls for eventual canonicalization [24, 25]. Because worms are imaged under a coverslip and thus lie in a 2D plane, such methods customarily model the 3D mapping as a 2D mapping, leaving the vertical dimension unchanged. We make the same simplification, because our current data does not provide sufficiently accurate 3D hull determination (worm edges are too blurred in less-focused horizontal slices far above or below the midplane).

Most previous methods do not attempt to preserve volume, and pose challenges related to distorted straightening at the head and tail. This is unfortunate, since these dense areas, presumably responsible for a majority of the worm's information processing, are the most important to get right for scientifically-relevant neuron identification. For example, we originally tested a simple 2D canonicalization where the new x-coordinate was defined as the distance along the worm midline estimated by skeletonizing the hull image, and the new y-coordinate was defined as the perpendicular distance to this midline. This scheme unfortunately resulted in problematic volume distortions associated with the contorted morphology of an unrestrained worm, and produced relatively poor neuron matching accuracy.

We therefore propose a worm canonicalization method which produces more consistent and biologically realistic point clouds. As explained below, the 2D mapping can be geometrically interpreted by filling the worm with inscribed circles as illustrated in Fig. 3. We first make a hull map for each worm, by binarizing the slice from the *C. elegans* *z*-stack that represents the largest-area 2D hull of the worm (Fig. 3, middle panel).

We then split the worm hull boundary into two parameterized curves representing opposite edges, $\mathbf{r}_-(s)$ and $\mathbf{r}_+(s')$, where $s \in [0, 1]$ and $\mathbf{r}_-(s)$ runs clockwise from head to tail as s increases. We define these curves by cubic spline fits to the binarized hull image. As illustrated in Fig. 3 (right panel), the distance from the point $\mathbf{r}_+(s')$ on one edge of the worm to a circle of radius t tangent to $\mathbf{r}_-(s)$ on the other edge is then given by

$$d(s, s', t) \equiv \left|\mathbf{r}_-(s) + \hat{\boldsymbol{n}}_-(s)t - \mathbf{r}_+(s')\right| - t, \tag{1}$$

where $\hat{\boldsymbol{n}}_-(s) \equiv (-\dot{r}_y(s), \dot{r}_x(s))/|\dot{\mathbf{r}}(s)|$ is the unit inward tangent vector at $\mathbf{r}_-(s)$, the vector $\mathbf{r}_-(s) + \hat{\boldsymbol{n}}_-(s)t$ is the center of the aforementioned circle, and dots denote derivatives with respect to s. By numerically minimizing over s', we find the distance $d(s, t)$ from the circle to the opposite worm edge:

$$d(s, t) \equiv \min_{s'} d(s, s', t). \tag{2}$$

We numerically solve the equation $d(s, t) = 0$ to determine the radius $t(s)$ where the circle is tangent to both worm edges. The variable $s \in [0, 1]$ thus parametrizes a continuous family of circles of radius $t(s)$ centered at

$$\mathbf{r}_0(s) \equiv \mathbf{r}_-(s) + \hat{\boldsymbol{n}}_-(s)t(s). \tag{3}$$

Finally, these circles are used to remap the *C. elegans* neurons using the methods illustrated in Fig. 3. For each worm, 1000 equally spaced circles are inscribed. All worm-inscribed circles are then translated so that their midpoints lie on the x-axis, retaining the distances between adjacent circle midpoints, and rotated so that the worm midline corresponds to the x-axis (Fig. 3, bottom). Each circle thus defines an affine transformation into the canonical space. Each neuron is mapped into this canonical space by applying the affine transformation for each of the circles that inscribe it, and averaging the result. Finally, the straightened group of neurons are isotropically rescaled so as to occupy $x \in [0\,\mu\text{m}, 800\,\mu\text{m}]$.

Atlas construction

After applying our canonicalization procedure to each individual worm and obtaining corresponding 3D neuron point clouds for each one, we combine these point clouds into a single NeuroPAL-derived whole worm atlas by using the median position coordinates for each neuron. We use the median rather than the mean since it is more robust to outliers and resistant to overall shrinking. To further improve the atlas quality in the neuron-dense head and tail regions, which are relatively rigid compared to highly deformable midbody, we rigidly align the high resolution head and tail atlases from [6] (which were based on 10 worms) to the two point clouds consisting of their corresponding neurons taken from our straightened 7-worm whole-body atlas, and finally combine these neuron positions with the original midbody positions to obtain our final atlas.

This resultant atlas is illustrated in Figs. 4 and 5, and the 300 canonicalized neuron position coordinates are listed in Additional file 1: S2 Appendix. We do not provide data for the two CAN cells in our dataset, as previous broad investigations of panneuronal markers found none that solely express in neurons and also express in CAN [26]. All panneuronal markers that expressed in CAN also expressed in non-neuronal tissues such as epithelium, intestine, glands, or muscle. Non-neuronal tissues are often larger than neurons and fluorescent expression within them can can occlude neuronal

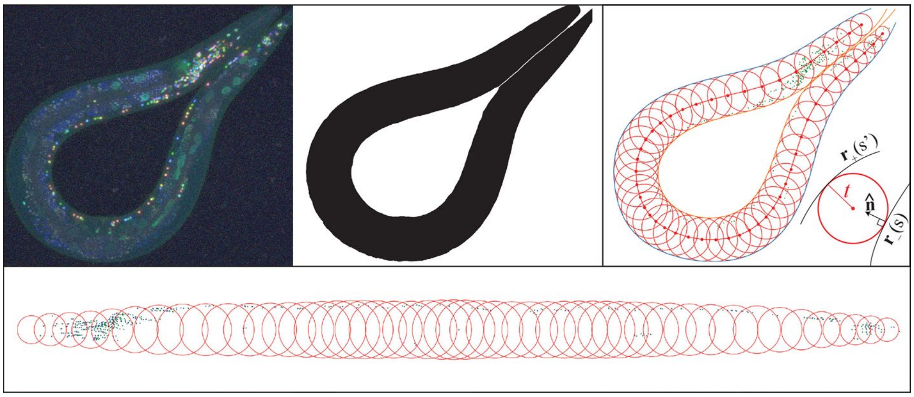

Fig. 3 Starting with a worm image (top left), we first determine the hull (top center). Our algorithm then inscribes 1,000 tangent circles (top right) in the hull (for clarity, only 50 circles are shown here). Finally, the circle sequence is straightened, which defines the canonicalization mapping of their inscribed neuron positions (bottom)

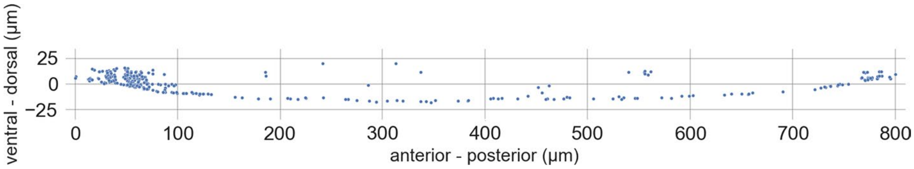

Fig. 4 Sagittal view of our NeuroPal-derived atlas

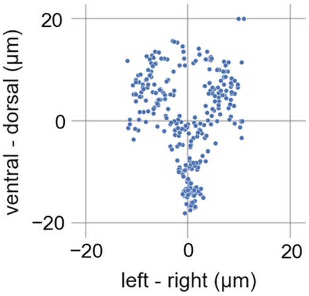

Fig. 5 Transverse view of our NeuroPal-derived atlas

imaging. Thus, to date, all whole-brain activity imaging as well as neural identification strains have used panneuronal markers that exclude CAN and, accordingly, these two cells are not represented in our dataset.

New alignment technique

We now turn to the challenge of aligning the neuron point cloud from an observed worm to a known atlas, so as to determine the likely identity of each observed neuron.

Existing coherent point drift alignment

Coherent Point Drift [27] (CPD) is a probabilistic alignment method for sparse point clouds that has been extensively used to align *C. elegans* neuron positions [5, 7, 19]. CPD represents the first point set by Gaussian mixture model (GMM) centroids and aligns to a second point set by maximizing the likelihood while forcing the points to move coherently as a group. In the non-rigid case, this is implemented as a motion over time guided by a velocity field determined by maximizing the GMM likelihood penalized by motion incoherence. CPD performs well on our *C. elegans* problem when the two point clouds undergo rigid-body rotation, translation and scaling, but as we will show below, has limited robustness in cases of cropping, size imbalance, and realistic biological deformation.

Colors

Since NeuroPAL is a fluorescent-labeling transgene that improves neuron distinguishability using several colors, it is highly desirable to exploit this color information to improve neuron matching. This begs the question: how many colors are needed for accurate alignment? With better alignment methods, are fewer colors sufficient? To quantify the relationship between color and accuracy, we report below a series of simulation results where each neuron is randomly assigned a simulated color.

Novel generalized-mean alignment

To improve the robustness of the alignment and address the issues in CPD, we introduce a novel *Generalized-Mean* (GM) alignment algorithm. For labeled neuron positions $\mathbf{r}^{(1)} = \{\mathbf{r}_i^{(1)}\}_{i=1}^{n_1}$ from reference worm 1 and unlabeled neuron positions $\mathbf{r}^{(2)} = \{\mathbf{r}_j^{(2)}\}_{j=1}^{n_2}$ from worm 2 (n_k denotes the number of neurons in the k^{th} worm), and optionally provided indexed colors $\mathbf{c}^{(1)} = \{c_i^{(1)}\}_{i=1}^{n_1}$ and $\mathbf{c}^{(2)} = \{c_j^{(2)}\}_{j=1}^{n_2}$, we introduce the following loss function to minimize by way of gradient descent on the $\mathbf{r}^{(2)}$ positions:

$$\ell_{GM}(\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \mathbf{c}^{(1)}, \mathbf{c}^{(2)}) = \sum_{j=1}^{n_2} \left(\frac{\sum_{i=1}^{n_1} \delta_{c_i^{(1)} c_j^{(2)}} \left| \mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)} \right|^{\gamma}}{\sum_{i=1}^{n_1} \delta_{c_i^{(1)} c_j^{(2)}}} \right)^{\frac{1}{\gamma}} \tag{4}$$

where $|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}|$ is the Euclidean distance between $\mathbf{r}_i^{(1)}$ and $\mathbf{r}_j^{(2)}$, and the Kronecker delta $\delta_{c_i^{(1)} c_j^{(2)}} = 1$ if the two colors are equal, vanishing otherwise. The generalized mean is defined by the hyperparameter $\gamma < 0$ which, as explained in [28], encourages pairing. More specifically, for each unlabeled neuron positions $\mathbf{r}_j^{(2)}$, we have a generalized-mean of its distance to all the labeled neuron positions $\mathbf{r}_i^{(1)}$ that have the same color (as enforced by $\delta_{c_i^{(1)} c_j^{(2)}}$). When $\gamma < 0$, the smaller the distance, the larger $\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}\right|^{\gamma}$ is. Furthermore, as proved in [28], this generalized-mean loss has the property that the smaller $\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}\right|$ is among all the $\mathbf{r}^{(1)}$, the larger the gradient is to drive them closer. Here γ tunes how much the algorithm is focusing on the smaller distances. For example, when $\gamma \to -\infty$, Eq. (4) reduces to $\sum_{j=1}^{n_2} \min_{\delta_{c_i^{(1)} c_j^{(2)}}=1} \left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}\right|$ where each unlabeled $\mathbf{r}_j^{(2)}$ only focuses on the nearest labeled $\mathbf{r}_i^{(1)}$ which can easily fall into a local minimum. However, allowing a *finite* negative γ also allows consideration of other potential parings that are not as near. In this paper, we set $\gamma = -6$, which we experimentally found to achieve a good balance between the aforementioned pairing effect and the desire for error correction whereby neurons at slightly larger distance produce enough of a gradient to be pushed together, which is crucial for avoiding getting trapped in suboptimal local minimal during the initial stages of training. In contrast, $\gamma = 2$ would correspond to ℓ being minus two times the log-likelihood of a Gaussian mixture model, making the loss similar to that of CPD with a Gaussian kernel.

Algorithm testing framework

To identify unknown neurons, our pipeline proceeds as illustrated in Fig. 2. It first canonicalizes the unlabeled neurons, then aligns them with the canonicalized atlas as described above. The alignment function takes as input an unlabeled point cloud of neuron positions, an atlas point cloud with known neuron IDs (with optional neuron colors), and returns transformed positions for the unlabeled point cloud (step 2). Finally, in step 3, each neuron with transformed position is assigned an ID by the following procedure: for the $n_2 \times n_1$ matrix of pairwise Euclidean distances between the n_2 neurons in the unlabeled point cloud and the n_1 neurons in the atlas, find the smallest element in the matrix, assign the corresponding ID, then delete this row and column in the matrix, and repeat. If colors are provided, each unlabeled neuron can only be assigned to a neuron in the atlas with the same color, so we choose the smallest element with unlabeled colors at each iteration.

We quantify the accuracy of all algorithms by testing them on simulated data where the ground truth is known. For these simulations, we use the OpenWorm dataset as the ground truth point cloud, distort it with simulated noise and biological deformations as described below, and finally measure which algorithms provide the most accurately reconstructed neuron identifications.

Parameterizing more realistic worm deformation

Working in the above-defined canonical space, we express the relation between the neuron positions $\mathbf{r}_i^{(1)}$ in reference worm 1 (our atlas, say) and observed neuron positions $\mathbf{r}_i^{(2)}$ for a worm 2 as

$$\mathbf{r}_i^{(2)} = \mathbf{r}_i^{(1)} + \mathbf{f}\left(\mathbf{r}_i^{(1)}\right), \tag{5}$$

for a *deformation function* $\mathbf{f}(r)$ that may include a random component. A very easy-to-simulate type of deformation is to simply add independent Gaussian noise to all coordinates of all neurons. This corresponds to treating all neuron positions $\mathbf{r}_j^{(2)}$ as independent parameters, and is tantamount to ignoring all biological constraints on tissue stretching.

We wish to regularize the problem to limit our analysis to more biologically plausible deformations, reflecting the known fact that positional variation between organisms exhibits correlation, whereby the deformation vector $\mathbf{f}$ is typically similar in direction and magnitude for adjacent neurons. In other words, we wish the deformation function $\mathbf{f}(\mathbf{r})$ to be relatively smooth, corresponding to nearby parts of the worm mainly shifting together as a coherent unit. We model the deformation field as a Gaussian mixture:

$$\mathbf{f}(\mathbf{r}) = \sum_{n=1}^{N} \mathbf{d}_n e^{-\frac{|\mathbf{r}-\mathbf{r}_n|^2}{2\sigma^2}} \tag{6}$$

Here the deformation function is parametrized by a vector $\mathbf{p}$ of $6N + 1$ parameters: the number σ, the components of the displacement vectors $\mathbf{d}_n$, and the displacement centers $\mathbf{r}_n$. A sample deformation field is visualized in Fig. 6. For better numerical stability, we add three redundant parameters in the form of an overall global displacement vector $\mathbf{s}$ added to $\mathbf{r}_n$. This deformation field is added as a term in our GM loss function, to create our "GM Realistic" algorithm.

Hyperparameter tuning

The performance of all tested alignment algorithms is highly hyperparameter dependent. To make a robust comparison of methods, we must ensure that for each

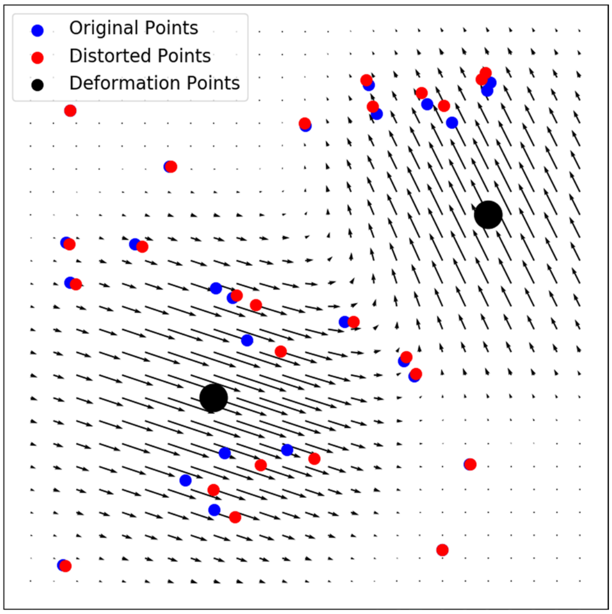

Fig. 6 Illustration of simulated biological noise simulated with the "bio-realistic" deformation field

algorithm we are using the optimal parameters for our use case. Therefore, we submit each algorithm to parallelized hyperparameter optimization.

Using the "Random Search" followed by the "Local Search" algorithms in the Sherpa [29] parameter tuning package, input parameters were tuned until alignment accuracy did not improve further. Here, alignment accuracy was defined as the average of performance per method in the tests "Robustness to Cropping", "Robustness to Dropout", and "Robustness to Biological Noise". The determined optimal parameters (Additional file 1: S1 Appendix) for each algorithm were used in all subsequent testing.

Results

We now compare the performance of our atlas and alignment methods with existing techniques. We designed experiments to answer the following questions: Firstly, how does our alignment method compare with existing methods against various types of adverse point cloud distortions? This is answered in Alignment performance on simulated data. Secondly, how does the new atlas perform compared to existing atlases? We answer this question in Testing alignment methods - Real Data by comparing the accuracy of aligning unlabeled neurons to existing and new atlases. Thirdly, since NeuroPAL and other labelling schemes use multiple colors to reveal identification, how many colors are necessary to achieve acceptable accuracy? In Testing alignment methods - Real Data, we also examine the effects of multiple colors on identification accuracy.

The following tests were devised: the point cloud positions of the 302 Openworm neurons were taken as a test set, as it can be assumed that these positions are reasonably representative of the morphology of actual *C. elegans* neurons [13]. Then, a copy of this point cloud was made, and selected perturbations, designed to simulate real world experimental conditions were performed. Finally, the group of points was randomly perturbed by a single vector drawn from a uniform distribution between ±5 microns in each of the *x*, *y*, and *z* directions. The alignment algorithms were then used to align the perturbed neuron point cloud to the original one. We test alignment with GM Realistic (GM with Gaussian deformation centers as defined in Eq. (6), to account for group neuron movement), CPD Rigid (CPD with rigid transformation, as defined in Fig. 2 of [27]), and CPD Deformable (CPD with deformable transformation, as defined in Fig. 4 of [27]).

This procedure was repeated 40 times at each perturbation setting, with resulting accuracies averaged together to provide a final metric of accuracy.

Alignment performance on simulated data

To simulate point clouds data similar in structure to those from real *C. elegans* images, we start with the OpenWorm atlas [13] and generate a point cloud of neuron positions with the ground-truth neuron IDs hidden from the alignment methods. A good alignment and identification pipeline should be robust to the above scenarios; therefore we sought to test the robustness of our alignment algorithm when deforming the Openworm point cloud dataset in these ways. In the following plots, results for optimally tuned parameters are shown.

Robustness to cropping

Commonly, *C. elegans* activity research focuses on the head of the worm, due to microscope field of view or resolution reasons, or hypotheses aimed at the nerve ring. During alignment, the head neurons are usually matched to a larger atlas. Therefore, a good identification method must perform well when aligning a dataset of unlabeled neurons that may reference a cropped subset of the comparison atlas.

For the first test, we start with a full Openworm point cloud that has been isotropically rescaled so as to occupy $x \in [-600\,\mu m, 200\,\mu m]$. We choose all head neurons (located at $x > 0\,\mu m$) as the unlabeled neuron set, and also create copies of the Openworm point cloud with various x cropping thresholds to act as the comparison atlas. We perturb the larger atlas as a rigid group away from the head neurons by a single vector drawn from a uniform distribution between ±5 microns in each of the x, y, and z directions. Then we use the alignment algorithms to align the unlabeled head neurons to the various croppings of the larger atlas, to test how different algorithms are robust to the imbalance of the two neuron sets.

Fig. 7 shows the resulting alignment accuracy for GM Realistic, CPD Rigid, and CPD Deformable. We can see that the GM Realistic algorithm performs perfectly across all levels of imbalance, while the CPD methods' accuracy drops quickly when the atlas extends beyond $-100\,\mu m$. The robustness of our GM method comes from the fact that the loss in Eq. (4) has the soft "clamping" effect that is able to focus on well-matched pairs of neurons while ignoring pairs that are matched poorly. The CPD methods, in contrast, are more sensitive to those badly-matched pairs, so the performance quickly drops when the two neuron sets are imbalanced. We also became aware of differences in alignment performance when aligning two sets of head neurons, from aligning two sets of whole-body neurons. The whole-body neurons take up space that is elongated along one axis, whereas head neurons occupy a relatively tight spherical volume. Because of this, head-head alignments were found to be more likely to fail by getting stuck in false local minima with the CPD methods.

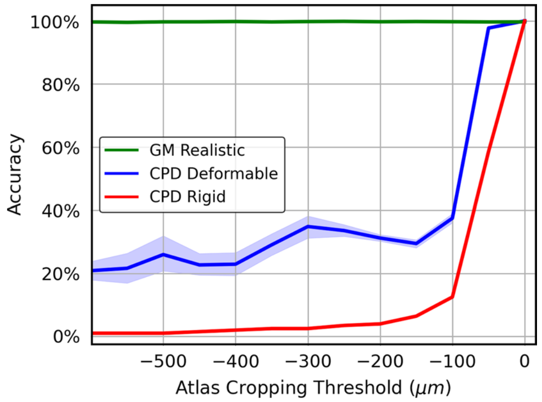

Fig. 7 Cropping Performance (40 run average): Alignment accuracy after aligning OpenWorm data cropped to the head to OpenWorm data cropped at various x-coordinates and perturbed as a group in a random direction in x, y, and z each at up to 5 microns. At $x = 0$, the atlas cropping isolates the head

Robustness to dropout

Fluorescent-protein expression that is used to identify neurons can at times be too dim to resolve. As a result of their dimness, these neurons cannot be detected or definitively identified. Such situations can occur due to many reasons, for example fast volumetric imaging methods that necessitate a tradeoff between speed and imaging quality. More generally, dimness is often found due to optical anisotropy present in various volumetric imaging techniques. A good identification method must therefore be robust to various levels of neural dropout.

For this, we start with a full OpenWorm point cloud. Then, we randomly remove a set number of neurons, and set the result as our unlabeled point cloud, with another copy of the OpenWorm point cloud as the comparison atlas. We perturb the larger atlas as a rigid group away from the head neurons by a single vector drawn from a uniform distibution between ±5 microns in each of the x, y, and z directions. Then we use the alignment algorithms to align the unlabeled set to the complete atlas.

Fig. 8 shows the resulting alignment accuracy for GM Realistic, CPD Rigid, and CPD Deformable. We can see that the GM Realistic algorithm performs the best in all cases of neuron removal, plateauing to 100% accuracy with only 80 out of 300 neurons remaining. The CPD methods are more prone to misidentification by getting stuck in suboptimal local minima.

Robustness to biological noise

Imaged *C. elegans* individuals do not typically exhibit identical morphology. For example, an individual that has consumed more food may be larger than another individual of the same age. These morphological differences are not isotropic, but their effects on distortion of adjacent neurons present as collective positional shift, so a good identification method must be robust to such local translation. We simulate such "biological noise" as follows.

We randomly generate N deformation points $\mathbf{r}_n$ within the volume of the neuron point cloud, and a corresponding displacement vector $\mathbf{d}_n$ drawn from a 3D Gaussian distribution of standard deviation σ, characterizing the "influence radius" of a deformation

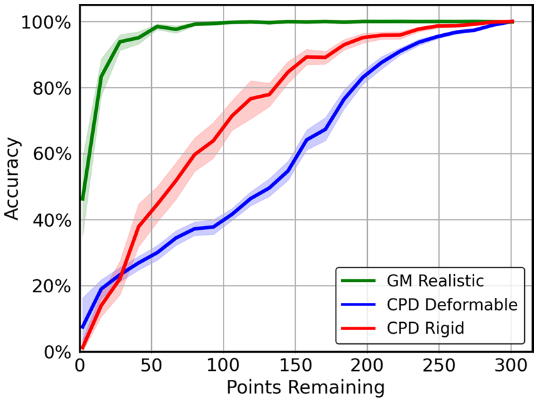

Fig. 8 Dropout Performance (40 run average): Alignment accuracy after aligning the OpenWorm atlas to the OpenWorm atlas with random points removed and perturbed as a group in a random direction in x, y, and z each at up to 5 microns

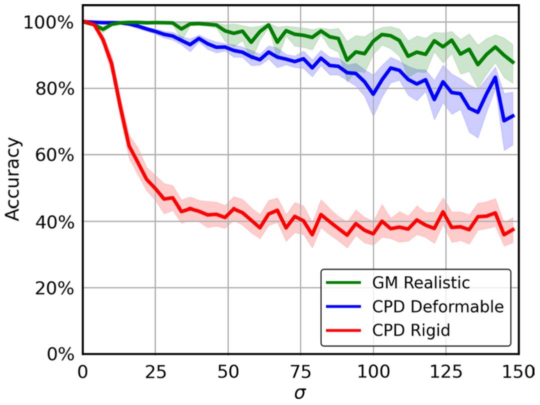

Fig. 9 Biological Noise Performance (40 run average): Alignment accuracy after aligning OpenWorm data to OpenWorm data that has been deformed with our biological noise method at various influence radius σ and perturbed as a group in a random direction in x, y, and z each at up to 5 microns

center, as illustrated in Fig. 6. In other words, each neuron is perturbed according to equation (6), with the only difference that now each $\mathbf{p}_n$ and $\mathbf{r}_n$ are randomly generated instead of learnable parameters.

This biological noise was applied to our test point cloud according to our realistic noise operation Parameterizing more realistic worm deformation, with $N = 100$ deformation centers, amplitude $|\mathbf{d}_n| = 6.1$ and varying influence radius $\boldsymbol{\sigma}$. We then perturb the larger atlas as a rigid group away from the head neurons by a single vector drawn from a uniform distribution between ±5 microns in each of the x, y, and z directions. Then we use the alignment algorithms to align the unlabeled set to the complete atlas.

In Fig. 9, we can see that the accuracy of CPD Rigid drops rapidly for biological noise influence radius beyond $15\,\mu\text{m}$. This may be because CPD Rigid assumes that all the neurons move as a rigid body that only allows rotation and translation as a whole, which is insufficient to model the realistic biological noise with relative expansion and contraction between neurons. On the other hand, while CPD Deformable performs much better than CPD Rigid, it underperforms GM Realistic across all the tested influence radii of the biological noise.

Testing alignment methods—real data

Atlas performance tests

We now test our NeuroPAL-based identification pipeline against previous atlas-based methods, in order to quantify their comparative performance, and also address the question of how many identification-assisting colors are necessary to achieve acceptable accuracy. Consider, for example, the case of aligning two point clouds, each consisting of the same 302 neurons. If every neuron in one point cloud was assigned the same unique distinguishable color as its corresponding neuron in the other (so that there are 302 distinct colors in all), any reasonable alignment algorithm exploiting color information should be able to produce 100% accurate alignment, since there is only one possible assignment for each neuron (its color twin). However, if we assigned 151 colors, each would correspond to two possible neuron identities. If we only assigned one single color to all neurons in each point cloud, an alignment algorithm would have 302 possible

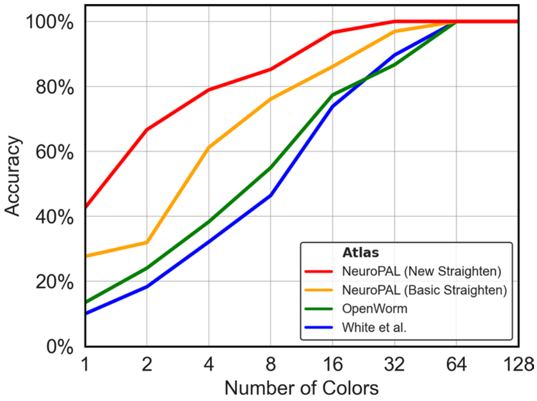

Fig. 10 CPD Rigid Alignment: CPD Rigid was used to align each of the 7 manually labeled NeuroPAL individual worm point clouds to the tested atlases, after cropping of the heads and tails. The figure shows how the median accuracy for these 7 worms depends on the number of unique colors used. For each number of colors $n > 1$, the colors were randomized and assigned to each neuron in both of the tested point clouds according to neuron identity, and CPD Rigid was run for each color such that it was only able to take into account the points that corresponded to the currently analyzed color

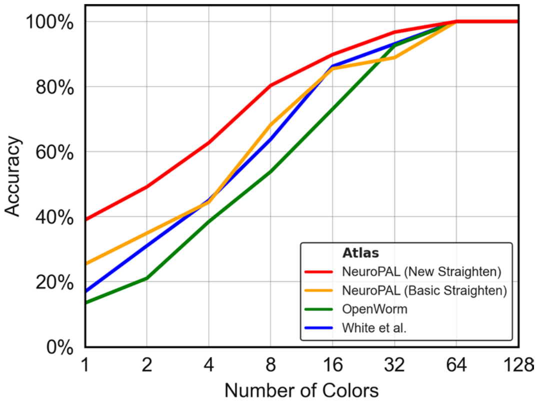

Fig. 11 CGM Realistic Alignment: Same as previous figure, but for CPD Deformable alignment

assignments for each neuron, and would have to rely on positional information alone. In Figs. 10, 11 and 12, we plot the performance of the three tested algorithms on our candidate atlases across a range of numbers of colors.

In addition, we include a version of the NeuroPAL atlas that has been straightened in a basic way as described by [24]. In these tests, all atlases were isotropically rescaled so as to occupy $x \in [0\,\mu\text{m}, 800\,\mu\text{m}]$, and to isolate the midbody the atlases were cropped at $x \in [200\,\mu\text{m}, 700\,\mu\text{m}]$. Then, alignments were performed using the same parameters as identified in Alignment performance on simulated data.

From the results, we have the following observations. Firstly, the NeuroPAL atlas outperforms the OpenWorm and White et al. atlases by a large margin, for all alignment methods, on mid-body data. This suggests that our NeuroPAL atlas combined with the circle straightening method enables more accurate identification. Secondly, comparing Figs. 12 and 10, we see that the GM Realistic method outperforms CPD methods across

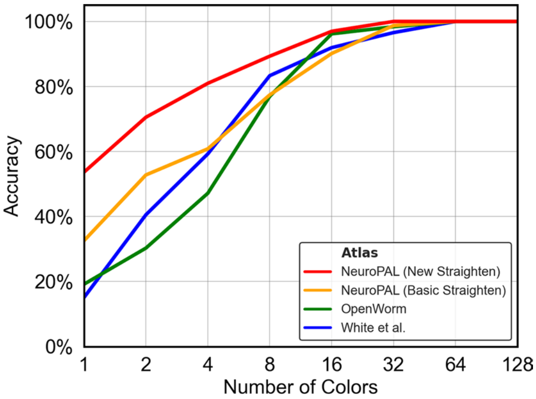

Fig. 12 GM Realistic Alignment: Same as previous figure, but for GM Realistic alignment. Here the accuracy for the 7 worms is computed by averaging 40 runs each, and then the median is taken as in the two previous figures. For each number of colors $n > 1$, the colors were randomized and assigned to each neuron in both of the tested point clouds according to neuron identity, and so GM Realistic now makes use of the c terms in Eq. (4)

all colors and atlases, again demonstrating that our alignment method allows more accurate neuron identification. Thirdly, as the number of colors increases, the performance of all atlases increases, emphasizing the important role that NeuroPAL can play in enabling more accurate automated neuron identification for atlas creation.

Conclusion

The main contributions of this paper are

1 a pipeline for constructing a 3D *C. elegans* atlas based on optically imaged neuron data,
2 an alignment method for identification of unlabeled *C. elegans* neurons using such an atlas, and
3 to the best of our knowledge, the most complete full-body *C. elegans* 3D positional neuron atlas, encapsulating positional variability derived from at least 7 animals per neuron.

We have presented tests suggesting that both our alignment algorithm and our pipeline-produced 3D atlas achieve higher identification accuracy than existing alternatives.

Many groups around the world are in the process of producing better imaging datasets so as to enable more promising investigation of myriad aspects of *C. elegans*, from neuronal activity to gene expression and cell-fate. We hope that, by delivering higher cell-type identification confidence, our atlas and others created using this method will help maximize the scientific value enabled by such functional imaging work.

Abbreviations

C. elegans: Caenorhabditis elegans; EM: Electron micrograph; L4: Fourth larval stage; nm: Nanometer; 2D: Two-dimensional; 3D: Three-dimensional; GFP: Green fluorescent protein; CFP: Cyan fluorescent protein; YFP: Yellow fluorescent protein; GMM: Gaussian mixture model; CPD: Coherent point drift; GM: Generalized-mean; ID: Identification; Eq: Equation.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04738-3.

Additional file 1. This supporting information includes the optimal hyperparameters found for the alignment algorithms, and a list of all neuron positions in our newly assembled whole-worm atlas.

Acknowledgements

The authors wish to thank Liam Paninski for helpful comments, and Soo Park for assistance with manual labelling.

Author contributions

MS, TW, EY, EB and MT wrote the main manuscript text. MS, TW and MT prepared the figures. MS and TW wrote and ran the alignment and testing code, and AN wrote code used for manual worm neuron labeling. EY prepared the samples, performed image acquisition and trained MS to manually label the neurons. MS, MT and EB conceived of the work. MS, TW, EY, AN, EB and MT reviewed the manuscript. All authors read and approved the final manuscript.

Funding

E.Y. was funded in part by the NIH (5T32DK7328-37, 5T32DK007328-35, 5T32MH015174-38, and 5T32MH015174-37). E.S.B. was supported by HHMI, Lisa Yang, John Doerr, and NSF Grant 1734870. M.T. was supported by the Foundational Questions Institute, the Rothberg Family Fund for Cognitive Science, and IAIFI through NSF Grant PHY-2019786.

Availability of supporting data

All code and supporting data is available at (https://github.com/bluevex/elegans-atlas) upon publication.

Declarations

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, USA. [2]Department of Computer Science, Stanford University, Stanford, USA. [3]Department of Neurobiology, University of Massachusetts Chan Medical School, Worcester, USA. [4]Department of Neuroscience, Columbia University, New York, USA. [5]Zuckerman Institute, Center for Theoretical Neuroscience, Columbia University, New York, USA. [6]Howard Hughes Medical Institute, Chevy Chase, USA. [7]Department of Physics, Institute for Brains, Minds and Machines, Institute for Artificial Intelligence and Fundamental Interactions, Massachusetts Institute of Technology, Cambridge, USA.

Received: 10 August 2021 Accepted: 17 May 2022
Published online: 28 May 2022

References

1. White J, Southgate E, Thomson JN, Brenner S. The structure of the nervous system of the nematode caenorhabditis elegans. Philos Trans Royal Soc Lond Ser B Biol Sci. 1986;314(1165):1–340.
2. Hall DH, Hartwieg E, Nguyen KC. Standard Immersion Fixation. https://www.wormatlas.org/EMmethods/Immersionfixation.htm
3. Hall DH, Hartwieg E, Nguyen KC. Modern electron microscopy methods for c elegans. In: Rothman JH, Singson A, editors. Methods in Cell Biology 2012;107: pp. 93–149. Elsevier
4. Zhang Y, Huang T, Jorgens DM, Nickerson A, Lin L-J, Pelz J, Gray JW, López CS, Nan X. Quantitating morphological changes in biological samples during scanning electron microscopy sample preparation with correlative super-resolution microscopy. PloS One. 2017;12(5):0176839.
5. Toyoshima Y, Wu S, Kanamori M, Sato H, Jang MS, Oe S, Murakami Y, Teramoto T, Park C, Iwasaki Y. Neuron id dataset facilitates neuronal annotation for whole-brain activity imaging of *c. elegans*. BMC Biol. 2020;18(1):1–20.
6. Yemini E, Lin A, Nejatbakhsh A, Varol E, Sun R, Mena GE, Samuel AD, Paninski L, Venkatachalam V, Hobert O. Neuropal: a multicolor atlas for whole-brain neuronal identification in *c. elegans*. Cell. 2021;184(1):272–88.
7. Scholz M, Linder AN, Randi F, Sharma AK, Yu X, Shaevitz JW, Leifer AM. Predicting natural behavior from whole-brain neural dynamics. bioRxiv 2018. https://doi.org/10.1101/445643. https://www.biorxiv.org/content/early/2018/10/17/445643.full.pdf
8. Choe Y, McCormick BH, Koh W. Network connectivity analysis on the temporally augmented *c. elegans* web: a pilot study. In: Society for Neuroscience Abstracts 2004;30.
9. Choe YC. Elegans Cell Location Data. https://github.com/yschoe/celegans
10. Kaiser M, Hilgetag CC. Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems. PLoS Comput Biol. 2006;2(7):95.
11. Hall DH, Altun ZF. *C. Elegans* Atlas. New York: Cold Spring Harbor Laboratory Press; 2007. p. 1.

12. Szigeti B, Gleeson P, Vella M, Khayrulin S, Palyanov A, Hokanson J, Currie M, Cantarelli M, Idili G, Larson S. Openworm: an open-science approach to modeling *caenorhabditis elegans*. Front Comput Neurosci. 2014;8:137.
13. Gleeson P, Lung D, Grosu R, Hasani R, Larson SD. c302: a multiscale framework for modelling the nervous system of *caenorhabditis elegans*. Philos Trans R Soc B Biol Sci. 2018;373(1758):20170379.
14. Varol E, Nejatbakhsh A, Sun R, Mena G, Yemini E, Hobert O, Paninski L. Statistical atlas of c. elegans neurons. In: International conference on medical image computing and computer-assisted intervention. Springer; 2020. p. 119–129.
15. Bubnis G, Ban S, DiFranco MD, Kato S. A probabilistic atlas for cell identification. arXiv preprint arXiv:1903.09227 2019.
16. Nejatbakhsh A, Varol E, Yemini E, Hobert O, Paninski L. Probabilistic joint segmentation and labeling of *c. elegans* neurons. In: International conference on medical image computing and computer-assisted intervention. Springer; 2020. p. 130–140.
17. Mena G, Nejatbakhsh A, Varo E, Niles-Weed J. Sinkhorn em: an expectation-maximization algorithm based on entropic optimal transport. arXiv preprint arXiv:2006.16548 2020.
18. Chaudhary S, Lee SA, Li Y, Patel DS, Lu H. Graphical-model framework for automated annotation of cell identities in dense cellular images. Elife. 2021;10:60321.
19. Yu X, Creamer MS, Randi F, Sharma AK, Linderman SW, Leifer AM. Fast deep learning correspondence for neuron tracking and identification in *c. elegans* using synthetic training. arXiv preprint arXiv:2101.08211 2021
20. Wen C, Miura T, Voleti V, Yamaguchi K, Tsutsumi M, Yamamoto K, Otomo K, Fujie Y, Teramoto T, Ishihara T. 3Dee-celltracker, a deep learning-based pipeline for segmenting and tracking cells in 3D time lapse images. Elife. 2021;10:59187.
21. Marblestone A. simple-C-elegans. https://github.com/adammarblestone/simple-C-elegans
22. Weissman TA, Pan YA. Brainbow: new resources and emerging biological applications for multicolor genetic labeling and analysis. Genetics. 2015;199(2):293–306. https://doi.org/10.1534/genetics.114.172510.
23. Tekieli T, Yemini E, Nejatbakhsh A, Varol E, Fernandez RW, Masoudi N, Paninski L, Hobert O. Visualizing the organization and differentiation of the male-specific nervous system of *c. elegans*. bioRxiv 2021.
24. Peng H, Long F, Liu X, Kim SK, Myers EW. Straightening *caenorhabditis elegans* images. Bioinformatics. 2008;24(2):234–42.
25. Christensen RP, Bokinsky A, Santella A, Wu Y, Marquina-Solis J, Guo M, Kovacevic I, Kumar A, Winter PW, Tashakkori N. Untwisting the *caenorhabditis elegans* embryo. Elife. 2015;4:10070.
26. Stefanakis N, Carrera I, Hobert O. Regulatory logic of pan-neuronal gene expression in *c. elegans*. Neuron. 2015;87(4):733–50.
27. Myronenko A, Song X. Point set registration: coherent point drift. IEEE Trans Pattern Anal Mach Intell. 2010;32(12):2262–75.
28. Wu T, Tegmark M. Toward an artificial intelligence physicist for unsupervised learning. Phys Rev E. 2019;100(3):033311.
29. Burke D, Laurino O, Wmclaugh Dtnguyen, Marie-Terrell Günther HM, Budynkiewicz J, Siemiginowska A, Aldcroft T, Deil C et al. sherpa/sherpa: Sherpa 4.12.1. Zenodo 2020. https://doi.org/10.5281/zenodo.3944985

Publisher's Note

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

Supporting information

S1 Appendix. **Hyperparameter settings.** Here, we provide the hyperparameters chosen for the GM Realistic and CPD Deformable algorithms, which have been optimized on the experiments in Robustness to Cropping and applied to all experiments. For GM Realistic, the exponent γ in Eq. (4) is set to -5. We train for $T = 5000$ epochs with learning rate $\mathrm{lr} = 6.31 \times 10^{-4}$ which is reduced by a factor of 0.3 if the best validation loss does not improve for 100 consecutive epochs (30 epochs for the experiment in Robustness to Cropping). To prevent numerical errors from division by near-zero during optimization when $\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}\right|$ is small (so that $\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}\right|^\gamma$ is large, since $\gamma < 0$), we clamp its value by replacing $\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)}\right|^\gamma$ with $\min\left(\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)} + \epsilon\right|^\gamma, 10^{20}\right)$ with $\epsilon = 10^{-10}$. Moreover, we add a small regularization term that encourages the unlabeled point cloud to keep its original shape, by regularizing Eq. (4) as follows:

$$\begin{aligned}
&\ell_{GM}(\{\mathbf{r}_i^{(1)}\}, \{\mathbf{r}_j^{(2)}\}, \{c_i^{(1)}\}, \{c_j^{(2)}\}) \\
&= \sum_{j=1}^{n_2} \left(\frac{\sum_{i=1}^{n_1} \delta_{c_i^{(1)} c_j^{(2)}} \min\left(\left|\mathbf{r}_i^{(1)} - \mathbf{r}_j^{(2)} + \epsilon\right|^\gamma, 10^{20}\right)}{\sum_{i=1}^{n_1} \delta_{c_i^{(1)} c_j^{(2)}}} \right)^{\frac{1}{\gamma}} + \lambda \left|\left|\mathbf{A}^{(1)} - \mathbf{A}_0^{(1)}\right|\right|_F^2
\end{aligned} \qquad (7)$$

Here $\mathbf{A}^{(1)}$ is the matrix containing pairwise distances between each neuron pair in $\mathbf{r}^{(1)}$, and $\mathbf{A}_0^{(1)}$ is $\mathbf{A}^{(1)}$ at the start of training. $||\cdot||_F$ denotes Frobenius norm. For λ, we use an annealing procedure of $\lambda(\tau) = \lambda_0 + (\lambda_m - \lambda_0) \cdot \max(1, \frac{\tau}{T \cdot \eta})$ where τ is the epoch number, and λ_0, λ_m, η are hyperparameters. In the experiment at Testing alignment methods - Real Data, we use $N = 200$ deformation centers, with $\sigma = 0.2$ and initial amplitude $|\mathbf{d}_0|$ of $\mathbf{d}$ being 0.015. In experiment at Robustness to Cropping, we use $N = 50$ deformation centers, with $\sigma = 50$ and initial amplitude $|\mathbf{d}_0|$ of 0.1, due to different scenario and units. The detailed hyperparameters that we tuned, including search range and the chosen value, is provided in Table 1.

Table 1 below provides the hyperparameter setting. The parameters α and β are defined in [25]. For other hyperparameters, we use the default setting as in its implementation at https://github.com/siavashk/pycpd (as of Aug 1, 2020).

Hyperparameter	Value	Search Range
γ	-5	{-1,-2,-3,-4,-5,-6,-7,-8}
lr	3.19×10^{-3}	$[1 \times 10^{-6}, 0.5]$
T	5000	{1000,5000}
λ_0	0.443019	[0, 1]
λ_m	0.241575	[0, 1]
η	0.766147	[0, 1]
α	0.0771160	[0, 10]
β	191.03671	[0, 200]

Table 1: Hyperparameters for the GM Realistic method (first group) and CPD Deformable method (α, β).

S2 Appendix. **The NeuroPAL atlas, tabulated.** Here, we provide the neuron coordinates for the NeuroPAL atlas that we generated over the course of this work. The coordinates of these 300 neurons can also be downloaded at `https://github.`

com/bluevex/elegans-atlas. As explained in the text, the two CAN cells are not included in our atlas.

Table 2: Positions of aggregated NeuroPAL atlas, in microns.

Neuron	x	y	z
ADAL	94.34	0.03	10.31
ADAR	91.22	-1.43	-11.64
ADEL	87.97	-0.12	10.50
ADER	83.23	-0.58	-11.47
ADFL	54.18	8.48	7.00
ADFR	54.63	7.15	-8.88
ADLL	59.13	12.05	3.86
ADLR	58.80	11.38	-6.34
AFDL	53.37	5.99	7.49
AFDR	53.11	5.71	-9.19
AIAL	66.74	-4.67	0.66
AIAR	67.33	-4.73	-0.99
AIBL	55.49	2.53	5.95
AIBR	55.94	2.61	-6.63
AIML	79.31	-6.77	2.87
AIMR	70.59	-5.01	-3.48
AINL	68.22	6.62	9.69
AINR	67.38	7.01	-10.84
AIYL	71.42	-4.40	5.34
AIYR	59.43	-1.49	-1.12
AIZL	68.92	2.53	7.98
AIZR	67.51	1.72	-8.68
ALA	51.26	15.36	-2.16
ALML	337.49	11.79	-11.86
ALMR	313.28	19.99	10.89
ALNL	784.36	12.09	3.51
ALNR	786.60	12.15	-4.00
AQR	87.59	-3.56	-10.65
AS1	125.47	-9.54	1.86
AS10	656.89	-9.97	-1.18
AS11	721.64	-5.19	0.47
AS2	178.62	-14.41	2.54
AS3	241.85	-13.12	1.37
AS4	314.15	-15.93	0.51
AS5	373.84	-16.51	0.11
AS6	416.50	-13.72	0.36
AS7	480.05	-12.86	1.70
AS8	535.33	-13.19	-0.11
AS9	598.95	-11.42	-0.09
ASEL	61.30	5.43	8.57
ASER	60.94	5.42	-10.37
ASGL	59.98	8.66	7.01
ASGR	60.15	8.00	-8.22
ASHL	58.42	4.21	8.35
ASHR	57.83	3.81	-10.20
ASIL	63.34	10.14	5.80
ASIR	63.42	9.72	-7.16
ASJL	64.06	-0.02	5.64
ASJR	64.01	-1.08	-5.99
ASKL	54.73	12.00	4.44
ASKR	54.43	11.18	-6.32
AUAL	60.91	0.84	8.36
AUAR	60.76	0.22	-9.63
AVAL	50.20	6.70	8.41
AVAR	50.99	4.79	-9.13
AVBL	63.74	5.10	7.12
AVBR	62.69	4.83	-9.41
AVDL	65.34	5.35	9.11
AVDR	65.45	5.14	-10.59
AVEL	51.08	4.58	5.98
AVER	52.03	3.63	-6.09
AVFL	94.37	-7.80	3.49
AVFR	95.54	-8.59	0.49
AVG	121.12	-9.44	-1.76

Table 2 – continued

Neuron	x	y	z
AVHL	62.68	9.65	7.85
AVHR	62.82	8.96	-9.77
AVJL	65.08	7.13	8.47
AVJR	65.13	7.55	-9.59
AVKL	87.39	-5.94	3.82
AVKR	75.68	-5.64	-3.17
AVL	70.82	-4.87	0.37
AVM	286.36	-1.16	10.16
AWAL	57.58	6.57	5.85
AWAR	57.24	6.53	-7.97
AWBL	57.36	8.30	8.52
AWBR	57.40	8.10	-10.25
AWCL	55.82	3.00	9.03
AWCR	56.43	1.74	-9.51
BAGL	38.69	5.69	8.54
BAGR	39.43	4.06	-8.24
BDUL	186.23	7.64	-10.13
BDUR	185.33	11.51	7.87
CEPDL	49.58	13.62	3.70
CEPDR	50.28	12.90	-7.07
CEPVL	39.34	1.03	4.58
CEPVR	40.68	1.01	-4.42
DA1	129.53	-9.48	-1.52
DA2	190.41	-14.38	-0.02
DA3	266.33	-14.92	-0.37
DA4	347.38	-17.97	-0.59
DA5	460.64	-14.18	1.56
DA6	546.48	-13.87	-0.03
DA7	656.94	-9.31	-0.37
DA8	745.92	-0.22	-1.35
DA9	743.47	-1.67	-2.51
DB1	121.18	-10.32	-1.02
DB2	99.22	-9.21	0.23
DB3	178.69	-14.50	-0.77
DB4	304.79	-16.26	-0.36
DB5	413.08	-14.18	0.87
DB6	532.94	-14.57	-0.14
DB7	650.43	-9.69	-0.78
DD1	124.58	-9.32	0.77
DD2	224.86	-13.53	1.26
DD3	351.57	-15.60	1.38
DD4	476.91	-14.26	2.44
DD5	591.74	-11.87	-0.19
DD6	736.31	-1.72	-0.59
DVA	770.84	10.96	-1.61
DVB	769.29	11.37	0.98
DVC	772.86	11.85	1.55
FLPL	81.11	0.02	10.58
FLPR	74.92	0.23	-11.36
HSNL	462.53	-1.78	-9.81
HSNR	452.01	-3.33	10.54
I1L	0.71	7.43	4.12
I1R	0.00	6.00	-2.71
I2L	13.08	4.85	7.23
I2R	13.82	3.39	-6.00
I3	18.92	13.57	-0.96
I4	71.62	10.84	-0.58
I5	72.18	-2.79	-2.56
I6	75.59	10.21	4.19
IL1DL	32.71	13.12	2.18
IL1DR	33.88	12.22	-4.59
IL1L	30.48	7.23	6.98
IL1R	30.97	5.64	-6.96
IL1VL	28.94	1.64	3.72
IL1VR	29.64	0.96	-1.72
IL2DL	24.21	12.34	4.34
IL2DR	26.43	12.45	-6.68
IL2L	29.90	7.93	9.15
IL2R	30.16	6.61	-8.80
IL2VL	25.35	2.14	5.06

Table 2 – continued

Neuron	x	y	z
IL2VR	28.36	0.63	-4.16
LUAL	774.32	6.28	5.16
LUAR	773.38	6.23	-5.35
M1	75.48	13.67	-6.40
M2L	69.41	2.97	3.49
M2R	69.57	4.17	-5.89
M3L	22.36	5.63	4.81
M3R	22.75	5.14	-2.91
M4	23.55	12.29	-0.69
M5	86.79	9.68	7.12
MCL	14.37	5.92	4.27
MCR	14.19	5.25	-3.51
MI	16.72	14.61	-1.03
NSML	15.78	4.53	7.54
NSMR	16.66	4.10	-5.46
OLLL	31.34	10.05	6.06
OLLR	32.56	8.59	-7.21
OLQDL	37.55	14.22	2.16
OLQDR	38.70	13.39	-5.63
OLQVL	36.66	3.93	6.16
OLQVR	37.26	3.11	-5.90
PDA	754.21	0.20	-2.03
PDB	748.26	-0.57	-3.40
PDEL	561.79	12.14	-7.53
PDER	555.40	11.49	10.46
PHAL	770.37	3.83	5.96
PHAR	770.06	4.14	-5.70
PHBL	773.56	5.55	6.00
PHBR	773.40	5.46	-6.33
PHCL	782.89	5.39	4.74
PHCR	782.27	5.49	-4.85
PLML	794.74	5.35	3.99
PLMR	795.65	5.12	-4.57
PLNL	770.37	3.83	5.29
PLNR	773.01	4.33	-4.55
PQR	776.96	4.86	5.20
PVCL	777.42	6.51	4.85
PVCR	776.14	5.98	-5.19
PVDL	555.97	12.66	-8.11
PVDR	540.27	11.60	7.00
PVM	556.10	9.96	-7.73
PVNL	794.35	6.90	3.24
PVNR	800.00	9.67	-2.87
PVPL	731.85	-3.27	1.07
PVPR	737.11	-1.81	-2.55
PVQL	766.78	6.02	6.41
PVQR	767.28	6.11	-5.59
PVR	786.71	8.07	-3.88
PVT	731.25	-2.90	-1.85
PVWL	782.97	7.52	4.18
PVWR	784.17	7.74	-4.13
RIAL	51.21	9.98	5.75
RIAR	51.66	9.13	-7.32
RIBL	62.04	3.07	6.02
RIBR	61.68	2.30	-6.76
RICL	70.85	0.28	8.03
RICR	69.97	-0.20	-8.23
RID	48.01	15.71	-2.99
RIFL	99.02	-8.14	2.31
RIFR	106.41	-9.05	-1.71
RIGL	114.67	-8.73	1.71
RIGR	117.71	-8.93	0.23
RIH	47.05	-0.69	-0.08
RIML	63.59	-0.63	4.45
RIMR	62.64	-1.99	-4.65
RIPL	35.30	7.97	6.37
RIPR	35.72	7.11	-6.56
RIR	54.10	0.76	1.40
RIS	87.93	-7.68	0.35
RIVL	64.03	11.92	4.43

Table 2 – continued

Neuron	x	y	z
RIVR	64.38	11.31	-7.21
RMDDL	55.78	0.20	3.91
RMDDR	56.06	-0.46	-2.32
RMDL	50.76	3.32	8.18
RMDR	52.31	1.87	-7.88
RMDVL	47.33	7.10	9.76
RMDVR	48.66	5.29	-10.21
RMED	40.14	15.50	-2.72
RMEL	38.03	8.71	7.36
RMER	38.76	7.16	-7.87
RMEV	44.66	-0.27	1.17
RMFL	58.12	-3.06	1.77
RMFR	57.51	-2.40	0.07
RMGL	98.07	-0.45	9.18
RMGR	96.44	-1.47	-10.33
RMHL	58.16	-0.78	3.15
RMHR	56.46	-1.22	-0.14
SAADL	53.44	-0.90	1.45
SAADR	50.27	-0.86	-0.39
SAAVL	49.05	9.73	6.88
SAAVR	49.34	8.24	-8.03
SABD	121.86	-9.41	-1.80
SABVL	86.35	-7.78	2.35
SABVR	80.61	-7.60	-2.17
SDQL	559.25	8.99	-7.45
SDQR	241.27	20.00	9.89
SIADL	55.68	-1.65	2.21
SIADR	53.27	-2.23	-1.59
SIAVL	66.79	-3.37	3.33
SIAVR	69.64	-4.92	-2.10
SIBDL	53.70	1.28	6.25
SIBDR	54.50	1.43	-6.72
SIBVL	59.54	-2.49	2.74
SIBVR	53.46	-1.56	-1.27
SMBDL	59.79	-1.85	2.44
SMBDR	60.22	-1.45	-1.90
SMBVL	62.73	-2.88	3.55
SMBVR	67.29	-4.46	-1.61
SMDDL	51.10	-0.48	2.46
SMDDR	54.10	0.01	-2.69
SMDVL	51.19	10.92	7.29
SMDVR	51.03	9.15	-9.28
URADL	30.69	12.53	4.50
URADR	31.32	11.24	-6.46
URAVL	33.57	2.35	4.74
URAVR	34.97	2.42	-3.85
URBL	34.05	7.53	8.41
URBR	34.27	6.21	-8.65
URXL	50.65	13.05	2.27
URXR	50.87	12.30	-5.18
URYDL	34.36	12.32	4.64
URYDR	35.22	10.80	-6.94
URYVL	32.70	5.12	6.71
URYVR	33.10	3.42	-6.05
VA1	108.33	-9.03	1.87
VA10	633.31	-10.82	-0.85
VA11	690.16	-7.21	0.31
VA12	737.02	-2.29	1.89
VA2	155.90	-12.66	0.30
VA3	209.48	-14.35	-0.15
VA4	274.69	-16.08	0.65
VA5	337.32	-16.96	1.41
VA6	407.98	-13.73	2.07
VA7	459.17	-14.93	2.93
VA8	505.74	-14.30	0.25
VA9	572.47	-13.18	-0.53
VB1	105.50	-8.78	1.39
VB10	581.55	-13.68	-0.70
VB11	640.50	-9.69	-0.66
VB2	85.76	-8.14	1.61

Table 2 – continued

Neuron	x	y	z
VB3	162.52	-13.05	0.50
VB4	217.31	-14.80	-0.10
VB5	287.47	-16.37	0.62
VB6	342.69	-16.78	1.46
VB7	405.49	-14.51	1.10
VB8	467.40	-15.05	2.99
VB9	524.97	-14.45	-0.04
VC1	224.54	-13.22	-0.14
VC2	294.26	-17.48	0.50
VC3	383.44	-16.60	1.40
VC4	441.76	-11.49	0.18
VC5	455.55	-8.49	0.58
VC6	530.44	-12.14	0.58
VD1	113.77	-8.15	1.06
VD10	602.94	-11.22	-0.04
VD11	660.86	-8.57	-0.53
VD12	727.56	-3.93	0.66
VD13	744.41	-0.35	-3.30
VD2	132.47	-9.95	-1.74
VD3	206.99	-13.59	-0.45
VD4	263.69	-14.60	1.18
VD5	319.93	-16.50	0.56
VD6	383.75	-15.85	0.96
VD7	431.61	-14.25	-1.05
VD8	482.54	-13.22	2.60
VD9	548.63	-13.48	-0.01