VNC-Dist: A machine learning-based pipeline for quantification of neuronal positioning in the ventral nerve cord of *C. elegans*

Saber Saharkhiz[1,3,4], Mearhyn Petite[3], Tony Roenspies[3], Theodore J. Perkins[2,4*], Antonio Colavita[1,3*]

Abstract

The ventral nerve cord (VNC) of newly hatched *C. elegans* contains 22 motoneurons organized into three distinct classes: DD, DA, and DB, that show stereotypical positioning and arrangement along its length. The VNC represents a genetically tractable model to investigate mechanisms involved in neuron sorting and positioning. However, accurately and efficiently mapping and quantifying all motoneuron positions within large datasets is a major challenge. Here, we introduce VNC-Dist (Ventral Nerve Cord Distances), a semi-automated software toolbox designed to overcome the limitations of subjective motoneuron positioning analysis in microscopy. The VNC-Dist pipeline integrates neuron localization using Fiji/ImageJ's Multi-point tool, automatic worm segmentation with the deep learning-based SAM-Plus, and neuronal distance measurement along the VNC via the VNC-Quant algorithm, which also supports statistical analysis and graphing. To demonstrate the robustness and versatility of VNC-Dist, we applied it to multiple genetic mutants known to disrupt neuron positioning in the VNC. This toolbox will enable the batch acquisition and analysis of large datasets on neuronal positioning per genotype, thereby advancing investigations into the cellular and molecular mechanisms that control neuron positioning and arrangement in the VNC. VNC-Dist will be publicly available: Streamlit Docker app, Python backend

1. Department of Cellular and Molecular Medicine, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
2. Department of Biochemistry, Microbiology and Immunology, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
3. Neuroscience, Ottawa Hospital Research Institute, Ottawa, ON, Canada
4. Regenerative Medicine, Ottawa Hospital Research Institute, Ottawa, ON, Canada
***Corresponding authors**: Antonio Colavita, colavita@uOttawa.ca; Theodore J. Perkins, tperkins@ohri.ca

Introduction

Fluorescent reporters have become invaluable tools for investigating nervous system connectivity and function in *C. elegans*[1]. The adult hermaphrodite has 302 neurons, organized into sensory neurons, interneurons, and motoneurons, with most cell bodies clustered in the nerve ring, retrovesicular ganglion (RVG), preanal ganglion (PAG), and along the ventral nerve cord (VNC)[2–4]. The position, morphology, and connectivity of different sets of these neurons can be readily visualized using distinct colours of cell-specific fluorescent markers[5]. However, accurately and reproducibly identifying and quantifying cells in densely packed neuronal regions remain challenging and time-consuming, particularly when analyzing large sample sizes or conducting high-throughput genetic screens[6–8].

At hatching, the *C. elegans* VNC contains 22 motoneurons: 6 DDs, 7 DBs, and 9 DAs, organized in a stereotyped pattern and distribution to control and coordinate muscle contractions along the entire length of the animal[9] (Fig. 1a,b). This arrangement is established during embryonic development through the directed movement and sorting of neuronal progenitors that form the VNC. Notably, this process shares underlying molecular and cellular mechanisms with neural tube formation in

vertebrates, suggesting that the VNC could serve as an anatomically simple and genetically accessible model to study the cell and molecular biology common to both[10]. Achieving this requires distinct labeling of DD, DA, and DB motoneurons, along with a workflow that efficiently measures the position of each neuron and the overall pattern of all 22 motoneurons along the VNC.

With the growing size and complexity of microscopy image datasets, faster and more efficient analytical workflows are essential[11]. Advancements in deep learning pipelines for bioimage analysis now enable researchers to analyze and interpret complex biological data more effectively[12,13]. Additionally, progress in biology is correlated with how fast and efficiently we can conduct experiments and analyses. Therefore, establishing dedicated bioimage analysis support using quantitative, reproducible methods within academic institutions is crucial for enhancing research quality and efficiency and significantly advancing scientific discovery in biology. Machine learning, particularly deep neural networks, enables predictive modeling through data-driven approaches that replicate human pattern recognition[14], making it ideal for automating the analysis of large and complex image datasets for more reproducible and efficient workflows[15–17]. In data-driven microscopy, computational methods optimize parameters like exposure time and illumination to enhance image quality, while machine learning algorithms handle tasks such as segmentation, classification, and quantification, facilitating high-throughput screening and rapid biological discoveries[18,19]. Integrating prior knowledge with data is crucial for developing informed machine learning models that address real-world challenges by pre-learning relevant patterns[20]. In the particular case of image segmentation, training datasets typically include raw images paired with corresponding labels using a supervised learning approach. However, this method is time-consuming, as both dataset creation and neural network training can take several hours to days[21]. Therefore, vision foundational models has the capability of zero-shot or few-shot learning—which enable models to perform segmentation of out-of-domain images—provide alternative approaches[22], such as Meta AI's Segment Anything Model (SAM) [23 23–25]. or its fine-tuned and enhanced versions for microscopy (μSAM)[26] and histopathology (PathoSAM)[27].

Accurate segmentation of objects in microscopy images remains a bottleneck for many researchers despite the number of tools developed for this purpose[26]. Current methods for quantifying curved biological structures often depend on traced path analysis. For example, contour-based methods (e.g., Neurite Tracer[28], SNT[29]), which estimate curve lengths by summing Euclidean distances between consecutive pixels, providing only a discrete approximation of the arc length. Pixel-based methods like skeletonization and NeuroQuantify[30] calculate lengths along the skeleton, while graph-based approaches such as NeuronJ represent pixels or regions as nodes linked by potential paths[30,31]. In contrast, Nellie's pipeline[32] uses hierarchical deconstruction and feature extraction to robustly quantify the complex architecture of organelles. However, many existing methods require extensive manual intervention and do not generalize well to models like C. elegans, forcing researchers to rely on time-consuming and less precise manual tools (e.g., the segmented line tool in Fiji/ImageJ; Fig. 2).

Visual detection of changes in VNC neuron positions is particularly challenging when comparing genetic mutants and assessing variability across individual animals. In wild-type first larval stage (L1) *C. elegans*, VNC neurons exhibit a consistent three-dimensional spatial distribution along the anterior–posterior (AP) axis (Fig. 1a). In contrast, in mutants, technical hurdles—such as distinguishing neurons from background noise, resolving densely packed neurons, and performing high-throughput analyses—complicate the reliable detection of subtle positional shifts[9] (Fig. S1a–e). Here, we present VNC-Dist (Ventral Nerve Cord Distances), a high-throughput, semi-automated pipeline specifically designed for labs studying *C. elegans*. VNC-Dist overcomes the challenges of distance measurement by quantifying neuron positioning along the VNC through the cumulative arc

length of the VNCL1 spline— a parametric curve that traces the VNC along the ventral boundary of the body[33] (Fig. 1b). This pipeline enables biologists to quickly and reliably analyze neuron positions from 3D fluorescent images using batch analysis, allowing for more accurate measurements while significantly reducing the time and effort compared to conventional manual methods. We demonstrate the pipeline's robustness and versatility by applying it to various genetic backgrounds, including *vang-1/Van Gogh*, *prkl-1/Prickle*, and *sax-3/Robo* m utants, that are known to cause VNC assembly defects, in both single and compound mutants[9]. This Python-based pipeline uses Meta AI's Segment Anything Model (SAM)[23,34] for automated worm segmentation, OpenCV[35] for shape extraction and contour detection, and a computational algorithm based on numerical analysis techniques such as Gaussian quadrature to measure distances between neurons[36].

1. Materials and Methods

1.1 Sample preparation

1.1.1 *C. elegans* strains, and image acquisition

Worms were cultured at 20°C. The N2 genetic background was used for all strains. We used the following alleles and transgenes organized by linkage group (LG): LGX: *vang-1(tm1422)* and *sax-3(zy5)*. LGII: *unc-4(zy123[unc-4::mNG])* and *zySi2[unc-30p::mcherry::H2B::unc-30]*. LGIII: *vab-7(zy137[vab-7::mNG::3xFlag])* and *vab-7(zy142[vab-7::mNG::T2A::mScarlet-I::H2B])*. LGIV: *prkl-1(ok3182)*.

All images of VNC neurons were acquired using a Zeiss Axio Imager M2 fluorescent widefield microscope equipped with a 40×/1.4 NA oil immersion objecti ve. Images were saved as .czi files in a four-dimensional format (CZYX: 2 channels, 11–17 z-slices, 1216 × 1920 pixels). The channels and z-stacks were combined using maximum intensity projection (MIP) and presented as 8-bit 2D images in .TIFF format. Imaging resolutions were optimized for neuron diameters—mScarlet (0.258 µm), mNG (0.225 µm), mCherry (0.266 µm)—with an overall merged image resolution of approximately 0.266 µm and an isotropic voxel size of 0.146 µm. During image acquisition, worms exhibiting twisting, hooking, or coiling were carefully excluded, as such deformations could compromise segmentation accuracy and data reliability. For sample preparation, newly hatched L1 stage worms were mounted on 2% agarose pad slides containing 0.1 µg/µL of the paralytic Levamisole in M9 buffer.

1.1.2 Construction of DD, DA, DB fluorescent reporter strains

zySi2 [unc-30p::mCherry::H2B::unc-30 3'UTR + unc-119(+)] was made using Mos1-mediated single-copy insertion (MosSCI)[37]. The MosSCI LGII ttTi5605 targeting and selection plasmid (pAC661) was constructed using Gibson Assembly to combine four PCR fragments: a ~2.5 kb *unc-30* promoter from N2 genomic DNA, a ~1.25 kb mCherry::H2B cassette from pOD2046 (Addgene #89367), a 410 bp *unc-30* 3'UTR from N2 genomic DNA, and a ~7.4 kb MOS targeting and selection cassette from pCFJ151. A sequence file for pAC661 is included in the supplemental materials.

CRISPR/Cas9 gene editing, utilizing the self-excising selection cassette (SEC) method, was used to knock-in mNG at the C-terminus of the endogenous *unc-4* and *vab-7* genes[38]. Guide sequences for C-terminal insertion were identified using the IDT guide design tool at

www.idtdna.com/site/order/designtool/index/CRISPR_SEQUENCE. Sequences were cloned into the Cas9–sgRNA plasmid pDD162 (Addgene #47549) using the NEB Q5 Site-Directed Mutagenesis Kit. *unc-4* and *vab-7* homology arms were cloned into the mNG-SEC plasmid pDD268 (Addgene #132523). Primers are listed in Supplemental Table S1.

A co-CRISPR approach[39] was used to swap the 3xFlag sequence in *vab-7(zy137[vab-7::mNG::3xFlag])* with a *T2A::mScarlet-I::H2B* cassette to generate *vab-7(zy142[vab-7::mNG::T2A::mScarlet-I::H2B]).* A homologous repair plasmid (pAC798) was made containing 5' and 3' homology arms PCR amplified from *zy132* and a *T2A::mScarlet-I::H2B* cassette ordered as an IDT gBlocks gene fragment. An sgRNA targeting a site within the 3xFlag sequence was generated using pDD162 as described[38]. Primers are listed in Supplemental Table S1 and a sequence file for pAC798 is included in the supplemental materials.

1.2 VNC-Dist pipeline

1.2.1 Pipeline design

We replaced the traditional heuristic approach—manually tracing neuron distances along the VNC with Fiji/ImageJ's Segmented Line tool (Fig. 2) for each of three motoneuron types, a process that is time-consuming, labor-intensive, and error-prone—with a more efficient strategy. We employed Fiji/ImageJ's Multi-point tool to annotate neurons in anterior-to-posterior order using counter numbers (1: DAs, SABs, Rectum; 2: DDs; 3: DBs). This method enabled us to extract the X-Y coordinates of all identified neurons, including the rectum, from the in-focus z-slices as a CSV file (Fig. 3a). We then merged and collapsed these in-focus slices into 2D TIFF images for segmentation by SAM-Plus. Finally, our pipeline computes the normalized relative distances along the AP axis of VNC by tracing ventral boundary of the body (VNCL1) (Fig. 1b). As shown in Fig. 3c, our pipeline significantly reduces processing time compared to manual analysis.

All components of VNC-Dist are implemented in Python, with core functions built upon widely recognized open-source libraries (i.e. SciPy[40], https://scipy.org/; NumPy[41], https://numpy.org/ for numerical analysis; scikit-image[42], https://scikit-image.org/; Pillow[43], https://python-pillow.org/; OpenCV[44], https://opencv.org/ for image processing; pandas[45], https://pandas.pydata.org/; Matplotlib[46], https://matplotlib.org/ for output visualization), whereas others were developed by us to perform specific tasks during the analysis. Then, to make the pipeline more accessible for biologists, we packaged VNC-Dist as a graphical user interface (GUI) using the Python-based Streamlit framework[47] (https://streamlit.io/), deployed within a Docker container on Linux/Ubuntu 24.04 LTS. This pipeline begins with a modified deep learning model, SAM-Plus, for segmenting 2D TIFF images of worms from noisy fluorescent micrographs. This model, comprising approximately 636.12 million parameters, operates on CUDA-enabled GPUs via the Compute Canada Graham cluster or dedicated GPUs (e.g., GeForce RTX) to batch-process all images. The subsequent CPU-based analysis can be efficiently performed on a standard local computer. The VNC-Dist pipeline includes two GUIs (Fig. 3b):

a) SAM-Plus (GPU-based) for worm segmentation as PNG masks.

b) VNC-Quant (CPU-based) for mapping the neurons along the VNCL1 (generating annotated PNG images) and calculating normalized relative distances as percentages (outputting as both individual and concatenated CSV files). It also provides different tabs for visualizing the data and statistical analysis.

Both GUIs launch in a web browser at separate ports (http://localhost:8502 for SAM-Plus and http://localhost:8501 for VNC-Dist) and can be run either via the command line or desktop shortcut. VNC-Dist is compatible with macOS, Linux, and Windows systems through Docker. Both SAM-Plus and VNC-Dist support batch processing, enabling rapid and efficient high-throughput data analysis.

1.2.1.1 Worm segmentation with SAM-Plus

We integrated the zero-shot promotable Segment Anything Model (SAM)[23] with the ViT-H image encoder into VNC-Dist without additional training (Fig. 4a). SAM provides two segmentation modes: automatic (grid-based) and interactive (user-prompted). To achieve full automation, we used the automatic mode.

For the automatic SAM mask generator, we configured it with 2 grid points per side to guide the attention towards worm, prediction IoU threshold of 0.98, stability score threshold of 0.98, crop layers of 1, downscale factor of 1 (no downsampling), and a minimum mask region area of 99,000 pixels to discard smaller regions. We processed batches of 2D merged and collapsed micrographs as input to SAM. Initial masks generated by SAM were refined by binarizing, inverting to highlight the foreground, and applying smoothing filters (Gaussian blur, median filter, bilateral filter), followed by morphological operations (erosion with a 19×19 kernel and closing with a 21×21 kernel). We created RGBA masks by setting the background to fully transparent, the segmented worm to opaque black, and adding a 25-pixel transparent border around the mask to enhance the representation of the segmented worm. The refined segmented worms were saved as .png files named to match their corresponding images. Due to the computational intensity of the process, an independent GUI version of the segmentation model, "SAM-Plus," was developed using a Docker container and the Streamlit Python package (Fig. 3b). This version is specifically optimized for execution on a dedicated GPU. Although imaging coiled worms is not recommended, separate tabs in the SAM-Plus GUI (Fig. 3b, tabs 2 and 3) allow for interactive correction of the segmented worms.

1.2.1.2 Contour detection and spline fitting with VNC-Quant algorithm

To automatically detect the outer boundaries of segmented binary worms and facilitate the fitting of the parametric cubic spline $S(t)$ to the generated contour points, we used OpenCV's contour detection algorithm as implemented in VNC-Quant. The algorithm organizes the detected contour points in a counterclockwise sequence, beginning from a reference point, typically the topmost-leftmost pixel (Fig. 4b). These ordered points were parameterized by t-values, which were used to construct cubic splines in the x and y dimensions. The resulting spline function, $S(t)$ provides a smooth and continuous representation of the worm's original shape for further analysis (Fig. 4c,d).

Within the VNC-Quant algorithm, the contour list C is rotated by an offset $\text{offset}=\lfloor N/2.31 \rfloor$, where N is the total number of contour points. This offset is determined based on the average size of the entire worm body contour at the L1 stage relative to the VNCL1 contour size on the ventral side. The starting index k is identified as the contour point $c_i \in C$ that is closest to projected SABVL (i.e., minimizing $\|ci-\text{SABVL}\|$). Applying the shift $k'=(k-\text{offset}) \bmod N$ ensures a smooth wrapping of the contour points without reordering neurons along the VNCL1 spline (Fig. 4b). Contour points and parametrization are defined as follows:

- Contour points (N points): $C = \{ (x_i, y_i) \in \mathbb{R}^2 \mid i = 1, 2, \ldots, N \}$
- Parametrization: $t_i = \dfrac{i-1}{N-1}, \quad i = 1, 2, \ldots, N$

This mapping ensures $t_1 = 0$ and $t_N = 1$, with uniformly spaced parameters in between.

Subsequently, we fit a parametric cubic spline $S(t) = (S_x(t), S_y(t))$ along the entire worm boundary to model its contours. Here, S_x: $[0,1] \to R$ approximates the x-coordinate, and S_y: $[0,1] \to R$ approximates the y-coordinate as functions of the parameter t. The equation below represents the regularized cubic spline fitting, with $S(0) = S(1)$ indicating that the worm's boundary is a closed curve. λ balances the trade-off between spline smoothness and data fit.

$$(S_x, S_y) = \arg \min_{\substack{S_x, S_y \\ \text{cubic splines}}} \left\{ \underbrace{\sum_{i=1}^{N} \left[(x_i - S_x(t_i))^2 + (y_i - S_y(t_i))^2 \right]}_{\text{data fidelity}} + \lambda \underbrace{\int_0^1 \left[(S_x''(t))^2 + (S_y''(t))^2 \right] dt}_{\text{smoothness penalty}} \right\}$$

Let t^*_{SABVL} denote the parameter corresponding to the projected SABVL onto the spline S(t):

$$t^*_{\text{SABVL}} = \arg\min_t \|\text{SABVL} - \mathbf{S}(t)\|^2$$

To measure the distance from projected SABVL (at t* SABVL) to any other projected neurons, we integrate along the VNCL1 spline, which is in the ventral side of S(t), using adaptive Gaussian quadrature (with u as a dummy variable of integration) (Fig. 4d):

$$L(t^*_{\text{SABVL}}, t) = \int_{t^*_{\text{SABVL}}}^{t} \sqrt{\left(\frac{dS_x}{dt}(u)\right)^2 + \left(\frac{dS_y}{dt}(u)\right)^2} \, \mathrm{d}u$$

Then, let MN = (MNx, MNy) represent the coordinates of a motoneuron. Its projection onto the spline S(t) is obtained by determining the parameter t* that minimizes the Euclidean distance between MN and S(t). Consequently, the projected motoneuron coordinates on the spline are given by S(t*) = (Sx(t*), Sy(t*)).

$$t^* = \arg\min_t \left\| \mathbf{MN} - \mathbf{S}(t) \right\|^2 = \arg\min_t \left(\left[S_x(t) - MN_x \right]^2 + \left[S_y(t) - MN_y \right]^2 \right)$$

Therefore, for a sequence of projected MNs along the VNCL1 with parameters $t_1, t_2, \ldots, t_N$, the arc length of the ith segment is

$$\Delta L_i = \int_{t_i}^{t_{i+1}} \sqrt{(\frac{dS_x}{dt}(u))^2 + (\frac{dS_y}{dt}(u))^2} \, du, \quad i = 1, \ldots, N-1$$

The cumulative arc length up to the nth point is then given by

$$C_n = \sum_{i=1}^{n-1} \Delta L_i, \quad \text{with } C_1 = 0$$

If L_{ref} denotes a reference arc length (the distance from projected SABVL to rectum), the normalized cumulative arc length is defined as

$$\tilde{C}_n = \frac{100\, C_n}{L_{\text{ref}}}$$

This normalization expresses each neuron position as a percentage of the reference length, referred to as the relative position.

1.3 Statistical analysis

All statistical analyses and graphing were performed in Python 3.10 using custom in-house scripts, which are also available through the user-friendly VNC-Dist GUI. To determine if WT neuron distributions were normal, we used a three-step approach: box–violin plot visualization, Q–Q plot analysis, and the Shapiro–Wilk test ($p = 0.05$) (Figs. S2–S4). Pairwise comparisons between WT and mutant groups were made using two-sample t-tests; depending on the outcome of Levene's F test, we applied either Student's t-test (equal variances) or Welch's t-test (unequal variances). Effect sizes were quantified using Cohen's d, considering results significant when $d \geq 0.7$ and $p < 0.05$. For WT, a correlation network analysis was conducted—using motoneurons as nodes and significant relative distances ($p = 0.05$, $r \geq 0.7$) as edges—to reveal the underlying neuronal pattern. Finally, due to the small sample size ($n = 6$), bootstrapping with replacement was employed to compare processing times between the manual method and the VNC-Dist pipeline.

2. Results

2.1 Multiplexed fluorescent labeling of DA, DD and DB motoneurons

To distinctly label the DD, DA, and DB motoneuron classes, we used their class-specific expression of transcription factors at the L1 stage: *unc-30* in DD1–6, *unc-4* in DA1–9 and the SAB neurons, and *vab-7* in DB1–7[48–50]. We therefore used a combination of red and green fluorophores to distinctly label the nuclei of each motoneuron class. MosSCI was used to insert a single copy of an *unc-30* promoter-driven *mCherry::H2B* transgene into the *ttTi5605* site on chromosome II. CRISPR/Cas9 gene editing was used to endogenously tag the C-terminus of *unc-4* with mNeonGreen and the C-terminus of *vab-7* with a dual-color tag, *mNeonGreen::T2A::mScarlet-I::H2B*. The T2A self-cleaving peptide[51] allows for co-expression of the mNG and mScarlet markers. mScarlet-I and mCherry were fused to histone H2B to target them to nuclei. In addition to DAs, *unc-4* is also expressed in the three SAB neurons, SABVL, SABVR, and SABD, located in the RVG[50]. When combined, this reporter system enables visualization of DD neurons in red, SAB and DA neurons in green, and DB neurons in yellow.

In newly hatched *C. elegans*, VNC motoneurons exhibit a stereotypical 3D spatial distribution along the AP axis. However, in genetic mutants such as *vang-1(tm1422)*, *prkl-1(ok3182)*, *sax-3(zy5)*, *vang-1(tm1422); prkl-1(ok3182)*, and *sax-3(zy5) vang-1(tm1422)* these patterns change in ways noy fully understood, resulting in inconsistent neuronal distributions within each genotype across different animals (Fig. S1).

2.2 An overview of the VNC-Dist workflow

VNC-Dist, a machine learning-based pipeline for quantification of VNC neuronal positioning, consists of two main parts. The first part employs GPU-accelerated batch processing to segment worms in micrographs using the state-of-the-art Segment Anything Model (SAM) from Meta AI (Fig. 4a). In the second part, we use CPU-based analysis to measure normalized relative distances of projected motoneurons along the VNCL1 spline, which is in the ventral side of $S(t)$, using SABVL

(anterior) and the rectum (posterior) as reference points (Fig. 4b, c). Establishing these consistent landmarks enables reliable genotype comparisons, as normalization accounts for the size variability among L1 worms.

To validate the relative distance measurements, the toolkit produces annotated images that display the projection of neurons onto the ventral side of the worm (VNCL1) along with the corresponding t-values for each neuron. This visualization confirms whether the ordering of t-values accurately reflects the anterior-to-posterior sequence of projected neurons according to the reference points (Fig. 4c, d). In addition, VNC-Dist supports batch processing. The process of our pipeline is as follows:

First, neuronal positions are determined using the Fiji/ImageJ Multi-point tool by manually selecting the centers of nuclei in 3D z-slices. Three distinct "Counter" numbers are assigned per class (1: DAs, SABs, and rectum; 2: DDs; 3: DBs), and the extracted X-Y coordinates are saved as .csv files (Fig. 4a).

Next, worm images are segmented using the SAM-Plus. The VNC-Quant algorithm then extracts the worm's contours and fits a smooth cubic spline, S(t). Neurons are projected onto the spline segment between the SABVL and rectum—termed VNCL1—and their positions are normalized as percentages of the arc-length (with SABVL at 0% and the rectum at 100%). This method accurately reflects the actual ventral nerve cord, which runs parallel to VNCL1(Fig. 4c,d).

2.3 Performance evaluation and key outputs from VNC-Dist

To evaluate the efficiency of the VNC-Dist pipeline, we applied it to WT, *vang-1*, *prkl-1* and *sax-3* single and double mutant images. The pipeline successfully segmented over 280 images using SAM-Plus and measured the relative distances of more than 7,000 neurons through VNC-Quant. Prior to conducting further analysis of VNC neuronal positioning to understand VNC assembly, we performed a sanity check by manually reviewing each annotated image to ensure that neurons were projected correctly, and their t-values were ordered accurately from anterior to posterior. Processing time comparisons using a custom timer Fiji macro demonstrated that VNC-Dist reduced analysis time by approximately 62% ($p=0.001$). In practice, while manual analysis of 50 images per genotype takes over 2.5 hours without interruption, VNC-Dist completes the task in about one hour (Fig. 3c).

As expected, the neuronal relative distances in WT exhibited normal distributions (Fig. 5a, b), as confirmed by the Shapiro-Wilk test ($p = 0.05$) and further supported by QQ-plot and box plot analyses (Figs. S2–S4). In addition, the analysis of neuron type arrangements revealed a consistent neuronal pattern across all animals (Fig.5c). For each neuron class (DA, DD, and DB), the mean sequential spacing pattern shows that motoneurons are arranged non-randomly and follow a specific organization along the anterior-posterior axis (Fig. 5d). On the other hand, the correlation network analysis revealed that all neurons along VNC could be categorized into three distinct clusters when normalized between SABVL (0%) and rectum (100%). The VNC neurons formed one cluster with minimal variation in the central region of the network, while the remaining two clusters comprised RVG and PAG neurons (Fig. 5e). These findings confirm that VNC neurons are anatomically distinct along the VNC.

To assess the impact of vang-1, prkl-1, and sax-3 mutations on neuronal positioning, we quantified the mean relative distance (with 95% confidence intervals) of neurons in the following strains: *sax-3(zy5)*, *sax-3(zy5) vang-1(tm1422)*, *prkl-1(ok3182)*, *vang-1(tm1422)*, and *vang-1(tm1422); prkl-1(ok3182)*, using our semi-automated pipeline. Analysis of neuron positions

revealed significant disruptions in the sorting patterns of neurons along the AP axis of VNC compared to WT worms.

Mutant strains exhibited significant anterior shifts in neuronal positions, with the double mutant *sax-3(zy5) vang-1(tm1422)* showing the most pronounced displacement (Fig. 6a). Furthermore, neuron type arrangement analysis of all 22 motoneurons (including 6 DDs, 7 DBs, and 9 DAs) across each genotype revealed neuron type switches along the AP axis of VNC (Fig. 6b). Also, some neuron types overlapped at a single position, indicating a potential intercalation issue along the z-axis (Fig. 6b). These disruptions in neuronal positioning were inconsistent within each mutant genotype. This suggests unpredictable disruptions in anterior neuronal positioning and underscoring the critical roles of these genes in maintaining proper neuronal positions. These findings align with our previous results from manual measurements, which focused solely on one type of motoneuron, DD, relative to DD1 using the Fiji/ImageJ Segmented Line tool [9].

2.4 Pitfalls and limitations

Our microscopy pipeline is not a generalist tool and is optimized for only neuronal positioning analysis along the AP axis of VNC at L1 stage.

It is recommended to refrain from imaging worms that exhibit twisting, hooking, or coiling. The efficacy of the segmentation process is highly contingent on the quality of Z-slice collapsing, which should be performed using MIP while selecting only the optimal in-focus slices.

3. Discussion

Here we introduced VNC-Dist, a semi-automated pipeline for quantifying neuronal positions within the VNC of *C. elegans* at L1, providing precise reproducible measurements crucial for understanding neuronal organization and genetic influence on positioning. This high-throughput interactive pipeline integrates deep learning with numerical analysis methods to robustly quantify neuron spacing along the AP axis of the VNC.

Our pipeline analyzed fluorescent micrographs from various mutant strains, including *sax-3*, *prkl-1*, *vang-1*, and their double mutant combinations, revealing distinct patterns of neuronal displacement. These findings underscore the critical roles of these genes in establishing the spatial order of motoneurons during VNC assembly. In addition, VNC-Dist confirmed that each motoneuron in the VNC of newly hatched wild type animals assumes a predictable position, validating the stereotypical pattern of neuronal spacing.

Our objective was to develop VNC-Dist to replace the time-consuming, labor-intensive, and error-prone manual distance measurements of VNC motoneurons. We envision further enhancement of VNC-Dist into a fully automated pipeline. Using state-of-the-art 3D cell and nuclei segmentation algorithms[52] , future iterations will aim to identify all motoneurons, including DA, DD, and DB types, thereby eliminating the need for manual annotation in Fiji/ImageJ to extract positional information. Furthermore, the segmentation component of VNC-Dist, SAM-Plus, does not require training dataset generation or model training; however, it can be fine-tuned using the mask decoder of the SAM algorithm with human-verified ground truth generated from SAM to facilitate the segmentation of even challenging coiled worms. Consequently, VNC-Dist offers a powerful platform that combines mathematical precision with machine learning to produce accurate results in a time-efficient manner.

Batch processing VNC neuron types from diverse fluorescent images and genotypes advances our understanding of the molecular and cellular mechanisms that govern neuronal positioning.

The toolkit's GUI and code will be open-sourced on GitHub after publication in a peer-reviewed journal.

Code availability

The reviewed version of the VNC-Dist pipeline is provided as Supplementary Software. The code, graphical user interface, and future updates are available at www.github.com/.... Additionally, test data, including TIF images and corresponding CSV coordinates, are provided to facilitate testing and evaluation of the pipeline.

Acknowledgements

Some strains were provided by the Caenorhabditis Genetics Center, which is funded by the National Institutes of Health Office of Research Infrastructure Programs (P40 OD010440). The authors acknowledge the Cell Biology and Image Acquisition Core (RRID: SCR_021845) funded by the University of Ottawa, Ottawa, Natural Sciences and Engineering Research Council of Canada (NSERC), and the Canada Foundation for Innovation. This work was supported by grants from the Canadian Institutes of Health Research (CIHR FRN 123513 and 156160) and NSERC (2018-06790) to A. Colavita, and an NSERC grant (2019-06604) to T. J. Perkins.

Declaration of interests

The authors declare no competing interests.

References

1. Hutter, H. Fluorescent protein methods: strategies and applications. in *Methods in cell biology* vol. 107 67–92 (Elsevier, 2012).
2. White, J. G., Southgate, E., Thomson, J. N. & Brenner, S. The structure of the ventral nerve cord of Caenorhabditis elegans. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences* **275**, 327–348 (1976).
3. Smith, J. J. *et al.* A molecular atlas of adult C. elegans motor neurons reveals ancient diversity delineated by conserved transcription factor codes. *Cell Rep* **43**, (2024).
4. Cook, S. J. *et al.* Whole-animal connectomes of both Caenorhabditis elegans sexes. *Nature* **571**, 63–71 (2019).
5. Taylor, S. R. *et al.* Molecular topography of an entire nervous system. *Cell* **184**, 4329–4347 (2021).
6. Sprague, D. Y. *et al.* Unifying community-wide whole-brain imaging datasets enables robust automated neuron identification and reveals determinants of neuron positioning in C. elegans. *bioRxiv* (2024).
7. Long, F., Peng, H., Liu, X., Kim, S. K. & Myers, E. A 3D digital atlas of C. elegans and its application to single-cell analyses. *Nat Methods* **6**, 667–672 (2009).
8. Hailstone, M. *et al.* CytoCensus, mapping cell identity and division in tissues and organs using machine learning. *Elife* **9**, e51085 (2020).
9. Shah, P. K. *et al.* PCP and SAX-3/Robo pathways cooperate to regulate convergent extension-based nerve cord assembly in C. elegans. *Dev Cell* **41**, 195–203 (2017).
10. Frith, T. J. R., Briscoe, J. & Boezio, G. L. M. From signalling to form: the coordination of neural tube patterning. (2023).
11. Saraiva, B. M. *et al.* Efficiently accelerated bioimage analysis with NanoPyx, a Liquid Engine-powered Python framework. *Nat Methods* 1–4 (2025).
12. Mahmud, M., Kaiser, M. S., McGinnity, T. M. & Hussain, A. Deep learning in mining biological data. *Cognit Comput* **13**, 1–33 (2021).
13. Greener, J. G., Kandathil, S. M., Moffat, L. & Jones, D. T. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology* vol. 23 Preprint at https://doi.org/10.1038/s41580-021-00407-0 (2022).
14. Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T. & Maida, A. Deep learning in spiking neural networks. *Neural networks* **111**, 47–63 (2019).
15. Moen, E. *et al.* Deep learning for cellular image analysis. *Nat Methods* **16**, 1233–1246 (2019).
16. Hallou, A., Yevick, H. G., Dumitrascu, B. & Uhlmann, V. Deep learning for bioimage analysis in developmental biology. *Development* **148**, dev199616 (2021).
17. von Chamier, L. *et al.* Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat Commun* **12**, 2276 (2021).
18. Morgado, L., Gómez-de-Mariscal, E., Heil, H. S. & Henriques, R. The Rise of Data-Driven Microscopy powered by Machine Learning. *arXiv preprint arXiv:2401.05282* (2024).
19. Jan, M., Spangaro, A., Lenartowicz, M. & Mattiazzi Usaj, M. From pixels to insights: Machine learning and deep learning for bioimage analysis. *BioEssays* **46**, 2300114 (2024).
20. Xu, H., Chen, Y. & Zhang, D. Worth of prior knowledge for enhancing deep learning. *Nexus* (2024).
21. Kalapos, A. & Gyires-Tóth, B. Self-supervised pretraining for 2d medical image segmentation. in *European Conference on Computer Vision* 472–484 (Springer, 2022).
22. Kadam, S. & Vaidya, V. Review and analysis of zero, one and few shot learning approaches. in *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1* 100–112 (Springer, 2020).
23. Kirillov, A. *et al.* Segment anything. in *Proceedings of the IEEE/CVF International Conference on Computer Vision* 4015–4026 (2023).
24. Ma, J. *et al.* Segment anything in medical images. *Nat Commun* **15**, 654 (2024).

25. Kirillov, A. *et al.* Segment anything. in *Proceedings of the IEEE/CVF International Conference on Computer Vision* 4015–4026 (2023).
26. Archit, A. *et al.* Segment Anything for Microscopy. *Nat Methods* (2025) doi:10.1038/s41592-024-02580-4.
27. Griebel, T., Archit, A. & Pape, C. Segment Anything for Histopathology. *arXiv preprint arXiv:2502.00408* (2025).
28. Longair, M. H., Baker, D. A. & Armstrong, J. D. Simple Neurite Tracer: open source software for reconstruction, visualization and analysis of neuronal processes. *Bioinformatics* **27**, 2453–2454 (2011).
29. Arshadi, C., Günther, U., Eddison, M., Harrington, K. I. S. & Ferreira, T. A. SNT: a unifying toolbox for quantification of neuronal anatomy. *Nat Methods* **18**, 374–377 (2021).
30. Dang, K. M. *et al.* NeuroQuantify–An image analysis software for detection and quantification of neuron cells and neurite lengths using deep learning. *J Neurosci Methods* **411**, 110273 (2024).
31. Meijering, E. *et al.* Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry A* **58**, 167–176 (2004).
32. Lefebvre, A. E. Y. T. *et al.* Nellie: automated organelle segmentation, tracking and hierarchical feature extraction in 2D/3D live-cell microscopy. *Nat Methods* 1–13 (2025).
33. Vincent, S. & Forsey, D. Fast and accurate parametric curve length computation. *Journal of graphics tools* **6**, 29–39 (2001).
34. Nikhila Ravi. SAM 2: Segment Anything in Images and Videos. *ArXiv* (2024).
35. Culjak, I., Abram, D., Pribanic, T., Dzapo, H. & Cifrek, M. A brief introduction to OpenCV. in *2012 proceedings of the 35th international convention MIPRO* 1725–1730 (IEEE, 2012).
36. Süli, E. & Mayers, D. F. *An Introduction to Numerical Analysis*. (Cambridge university press, 2003).
37. Frøkjær-Jensen, C. *et al.* Single-copy insertion of transgenes in Caenorhabditis elegans. *Nat Genet* **40**, 1375–1383 (2008).
38. Dickinson, D. J., Pani, A. M., Heppert, J. K., Higgins, C. D. & Goldstein, B. Streamlined genome engineering with a self-excising drug selection cassette. *Genetics* **200**, 1035–1049 (2015).
39. Arribere, J. A. *et al.* Efficient marker-free recovery of custom genetic modifications with CRISPR/Cas9 in Caenorhabditis elegans. *Genetics* **198**, 837–846 (2014).
40. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* **17**, 261–272 (2020).
41. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
42. Van der Walt, S. *et al.* scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014).
43. Clark, A. Pillow (pil fork) documentation. *readthedocs* (2015).
44. Mordvintsev, A. & Abid, K. Opencv-python tutorials documentation. *Obtenido de https://media. readthedocs. org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals. pdf* (2014).
45. McKinney, W. Data structures for statistical computing in Python. in *SciPy* vol. 445 51–56 (2010).
46. Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput Sci Eng* **9**, 90–95 (2007).
47. Khorasani, M., Abdou, M. & Hernández Fernández, J. Web Application Development with Streamlit. *Software Development* 498–507 (2022).
48. Esmaeili, B., Ross, J. M., Neades, C., Miller III, D. M. & Ahringer, J. The C. elegans even-skipped homologue, vab-7, specifies DB motoneurone identity and axon trajectory. (2002).
49. Jin, Y., Hoskins, R. & Horvitz, H. R. Control of type-D GABAergic neuron differentiation by C. elegans UNC-30 homeodomain protein. *Nature* **372**, 780–783 (1994).
50. Miller III, D. M. & Niemeyer, C. J. Expression of the unc-4 homeoprotein in Caenorhabditis elegans motor neurons specifies presynaptic input. *Development* **121**, 2877–2886 (1995).
51. Liu, Z. *et al.* Systematic comparison of 2A peptides for cloning multi-genes in a polycistronic vector. *Sci Rep* **7**, 2193 (2017).
52. Ma, J. *et al.* The multimodality cell segmentation challenge: toward universal solutions. *Nat Methods* 1–11 (2024).

VNC-Dist: A machine learning-based pipeline for quantification of neuron positioning in the ventral nerve cord of *C. elegans*

Saber Saharkhiz[1,3,4], Mearhyn Petite[3], Tony Roenspies[3], Theodore J. Perkins[2,4*], Antonio Colavita[1,3*]

1. Department of Cellular and Molecular Medicine, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
2. Department of Biochemistry, Microbiology and Immunology, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
3. Neuroscience, Ottawa Hospital Research Institute, Ottawa, ON, Canada
4. Regenerative Medicine, Ottawa Hospital Research Institute, Ottawa, ON, Canada

***Corresponding authors**: Antonio Colavita, colavita@uOttawa.ca; Theodore J. Perkins, tperkins@ohri.ca

Figures and Legends

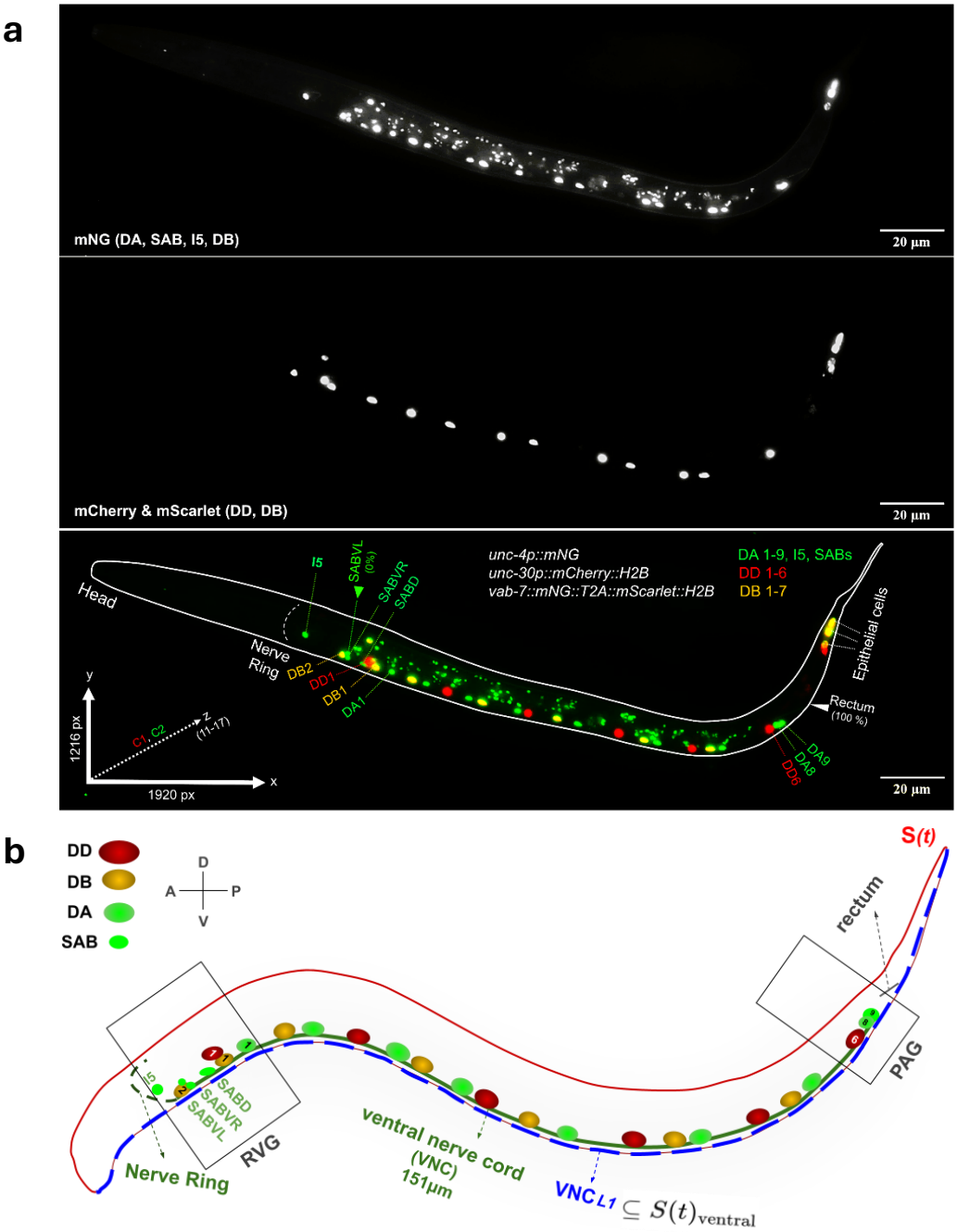


Fig. 1 | Nervous system of a newly hatched wild type *C. elegans* hermaphrodite

(a) Fluorescent micrograph of the VNC expressing nuclear-targeted fluorescent reporters. In our triple reporter strain, mCherry labels DD neurons (red), mNG labels DA, SAB and I5 neurons (green), and the overlapping expression of mScarlet and mNG labels DB neurons (orange). In addition, epithelial cells in the tail region are labeled with the DB reporter. Channels were merged and optimal z-slices were collapsed through maximum intensity projection (MIP) for displaying all motoneurons along the anterior–posterior (AP) axis. Reference points are set at SABVL (0% anterior) and the rectum (100% posterior). For clarity, single-channel images for green and red are also provided. Scale bar, 20 µm (1 µm = 6.82 pixels). **(b)** Diagram of a wild type (WT) *C. elegans* at L1 highlighting key nervous system structures: the nerve ring, ventral nerve cord (VNC), motoneurons, SAB neurons (SABVL, SABVR, SABD), and the I5 pharyngeal neuron. The retrovesicular ganglion (RVG) includes SAB neurons, DA1, DB1, DB2, and DD1; the preanal ganglion (PAG) comprises DA8, DA9, and DD6. The worm's body is modeled as a parametric spline function S(t) (red), with the VNCL1 spline (dashed blue) running along the ventral side of S(t), depicted in parallel to the VNC (green).

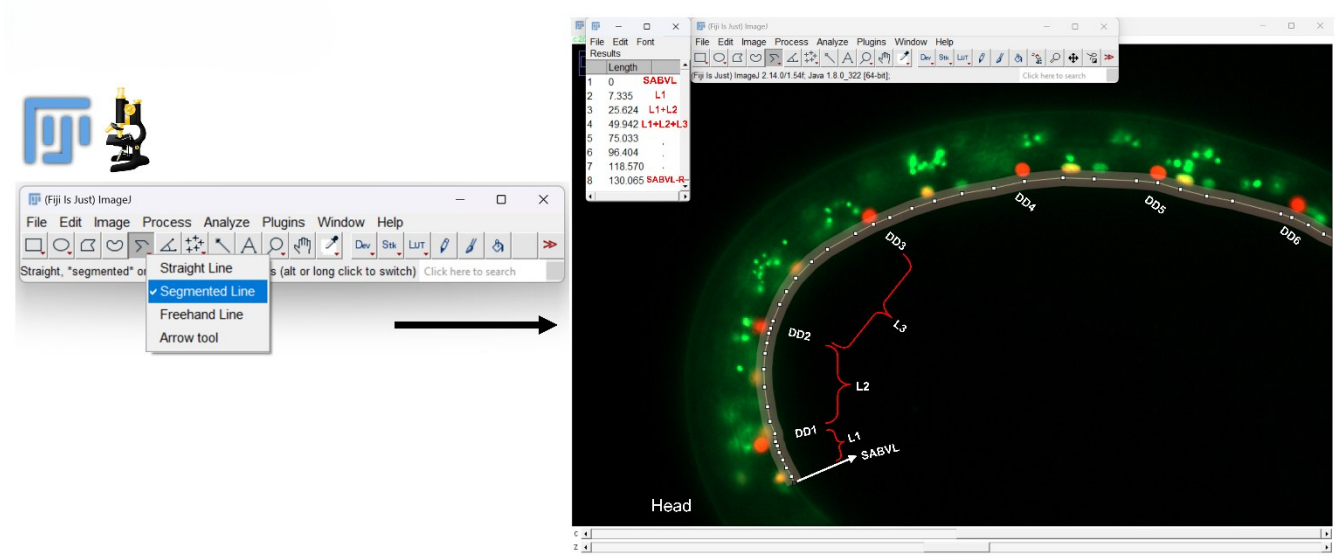
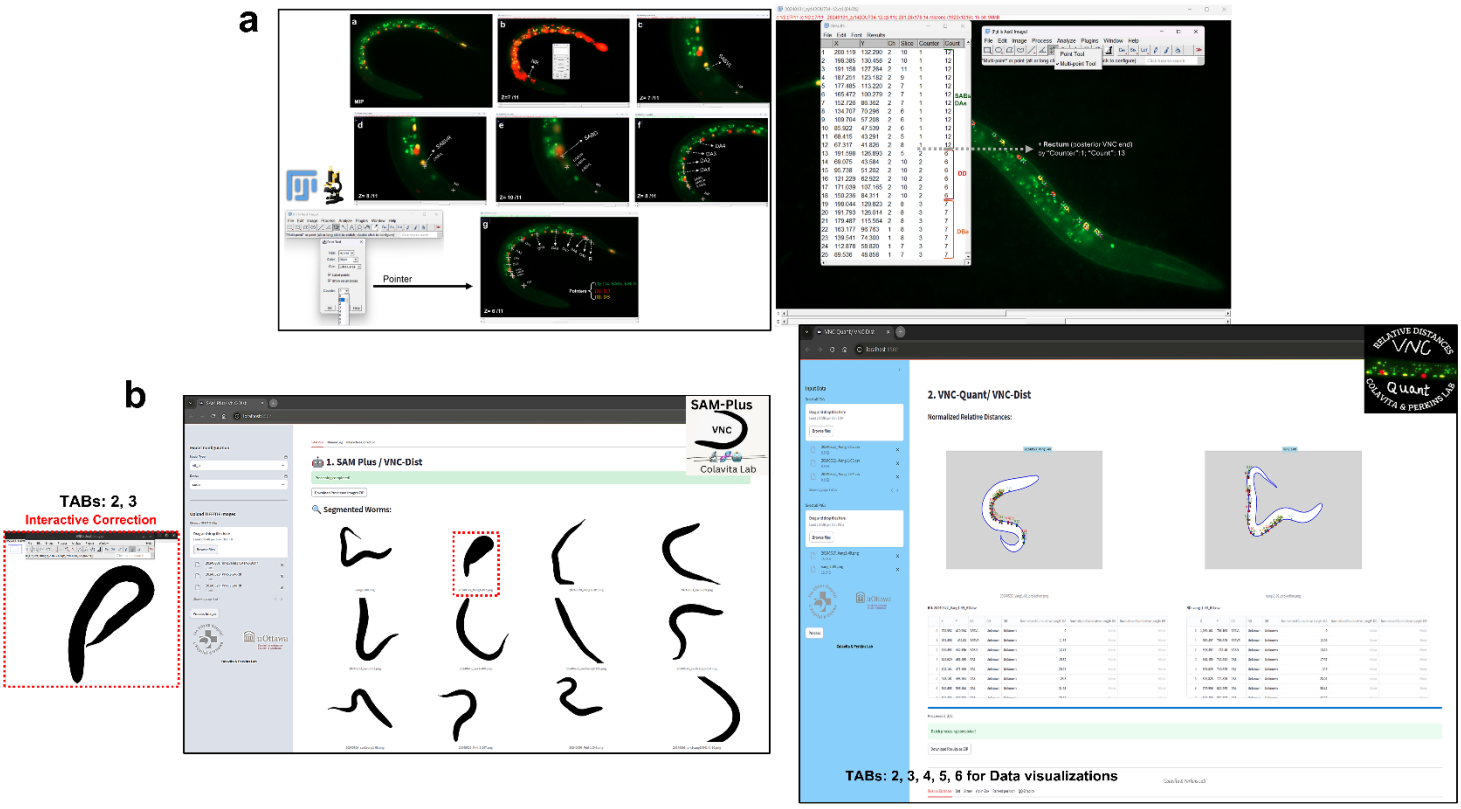


Fig. 2 | Heuristic method using Fiji/ImageJ for scoring and measuring neuron positioning distances

Manual measurement of relative distances performed with the Segmented Line tool in Fiji/ImageJ. Distance analysis involved three separate traversals along the ventral nerve cord (VNC) for each of the 22 neurons per class: Green neurons (DAs, SABs), Red neurons (DDs), and Orange neurons (DBs).



c

Method	Time
Manual	3.05 min ± 23.81 s
Fiji Annotator in VNC-Dist	1.15 min ± 5.08 s

***

Fig. 3 | User interface of VNC-Dist pipeline for analyzing VNC motoneuronal positioning

(a) Neurons from SABVL to the Rectum were annotated using Fiji/ImageJ Annotator (Multi-Point tool) and classified into three groups based on Counter numbers (1: DAs, SABs, Rectum; 2: DDs; 3: DBs). The X-Y coordinates were saved as a CSV file, and in-focus z-slices were merged and collapsed into 2D TIFF images. **(b)** The left displays the SAM-Plus GUI for CUDA GPU-based batch segmentation of representative worms, with results exportable as a zip file (Tab1) and interactive correction/manual segmentation for coiled worms (Tabs 2 and 3). The right shows the VNC-Quant GUI, which maps normalized motoneuron positions along the VNC's anteroposterior axis (Tab1) and generates a spreadsheet of neuron positions (DDs, DBs, DAs, SABs) per genotype, with Tabs 2–6 offering various data visualizations. **(c)** Processing times, recorded by a custom Fiji timer macro, were compared for six images per genotype (n = 6). VNC-Dist significantly reduced processing time by around 62% relative to the manual method (mean ± SD from 500 bootstrap samples; ***p = 0.001).

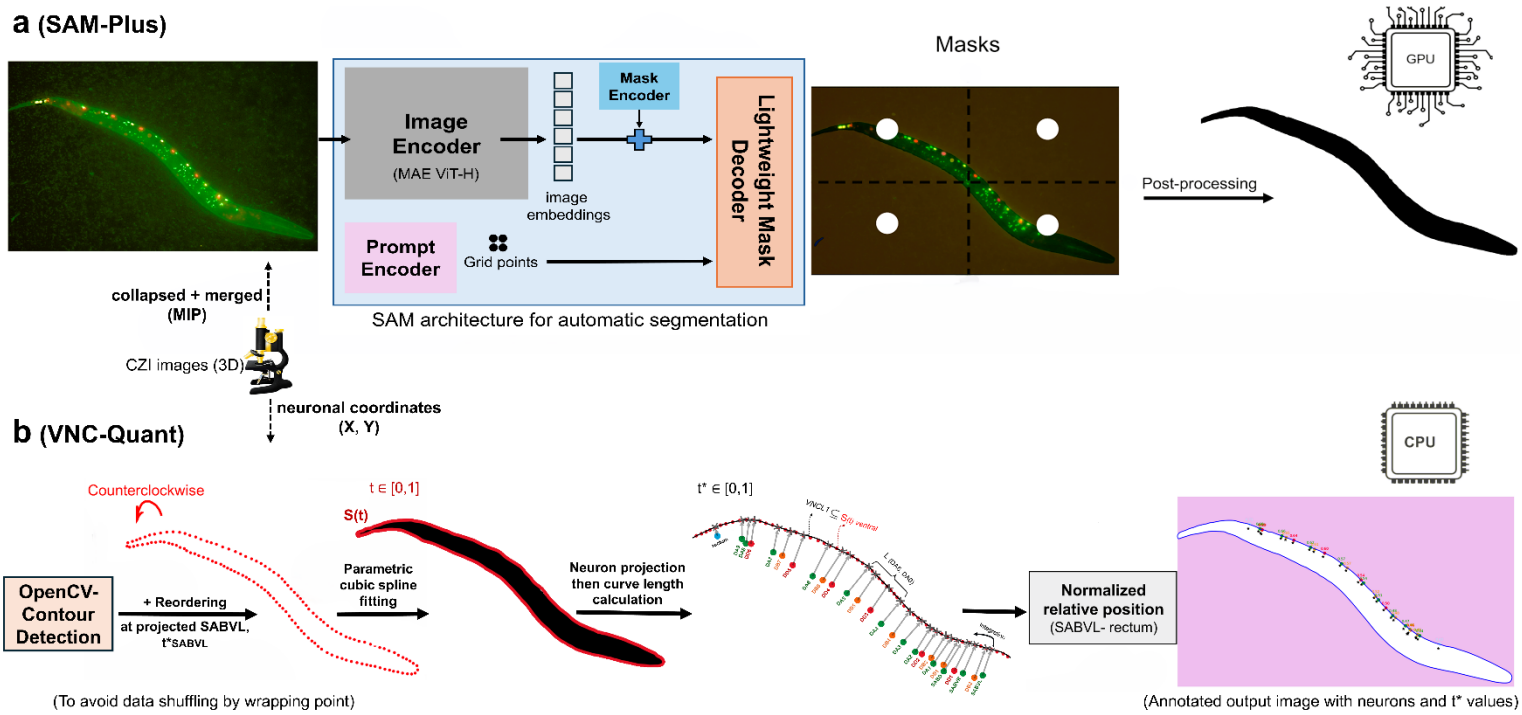


Fig. 4 | Overview of VNC-Dist pipeline for analyzing VNC neuron positioning.

CZI images are first compressed and merged, and neuron coordinates are extracted in Fiji/ImageJ. These data are then processed with VNC-Dist in two steps: **(a)** SAM-Plus automatically segments the worm body using the Segment Anything Model (SAM), prompted by a grid of two points per side, followed by post-processing to isolate and enhance the worm contour; and **(b)** VNC-Quant uses OpenCV to extract the pixel-level worm boundary, fits a parametric cubic spline S(t) to these points, and projects each neuron onto this spline (VNCL1). Neuron positions are calculated via adaptive Gaussian quadrature, from the anterior (projected SABVL, 0%) to the posterior (rectum, 100%).

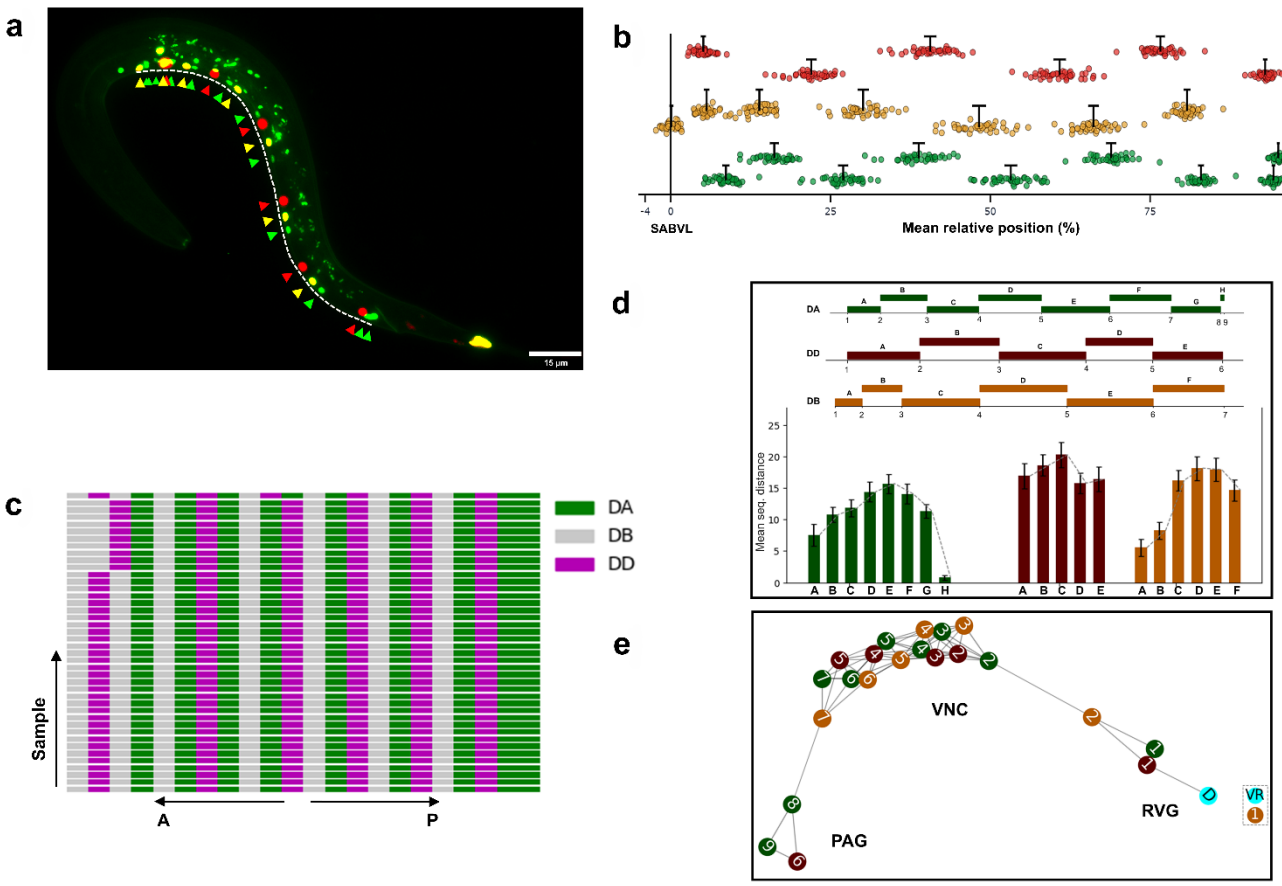


Fig. 5 | VNC Neuron positioning in wild type *C. elegans* analyzed by VNC-Dist pipeline

(a) Merged and collapsed micrograph showing motoneurons along the AP axis (dashed line). Green arrows: DAs and SABs; red arrows: DDs; orange arrows: DBs. Scale bar, 15 µm. **(b)** Dot plot of normalized relative distance of motoneurons from SABVL (0%) to rectum (100%) for n = 42. Error bars represent mean ± 95% CI. **(c)** Neuron type arrangement analysis of all 22 motoneurons from anterior to posterior. **(d)** Mean sequential spacing of neurons along the AP axis, with segmented bar plots (top). Error bars represent mean ± SD. **(e)** Correlation network analysis using motoneurons as nodes and significant relative distances ($p = 0.05$, $r \geq 0.7$) as edges to group neurons into VNC, RVG, and PAG relative to SABVL (0%).

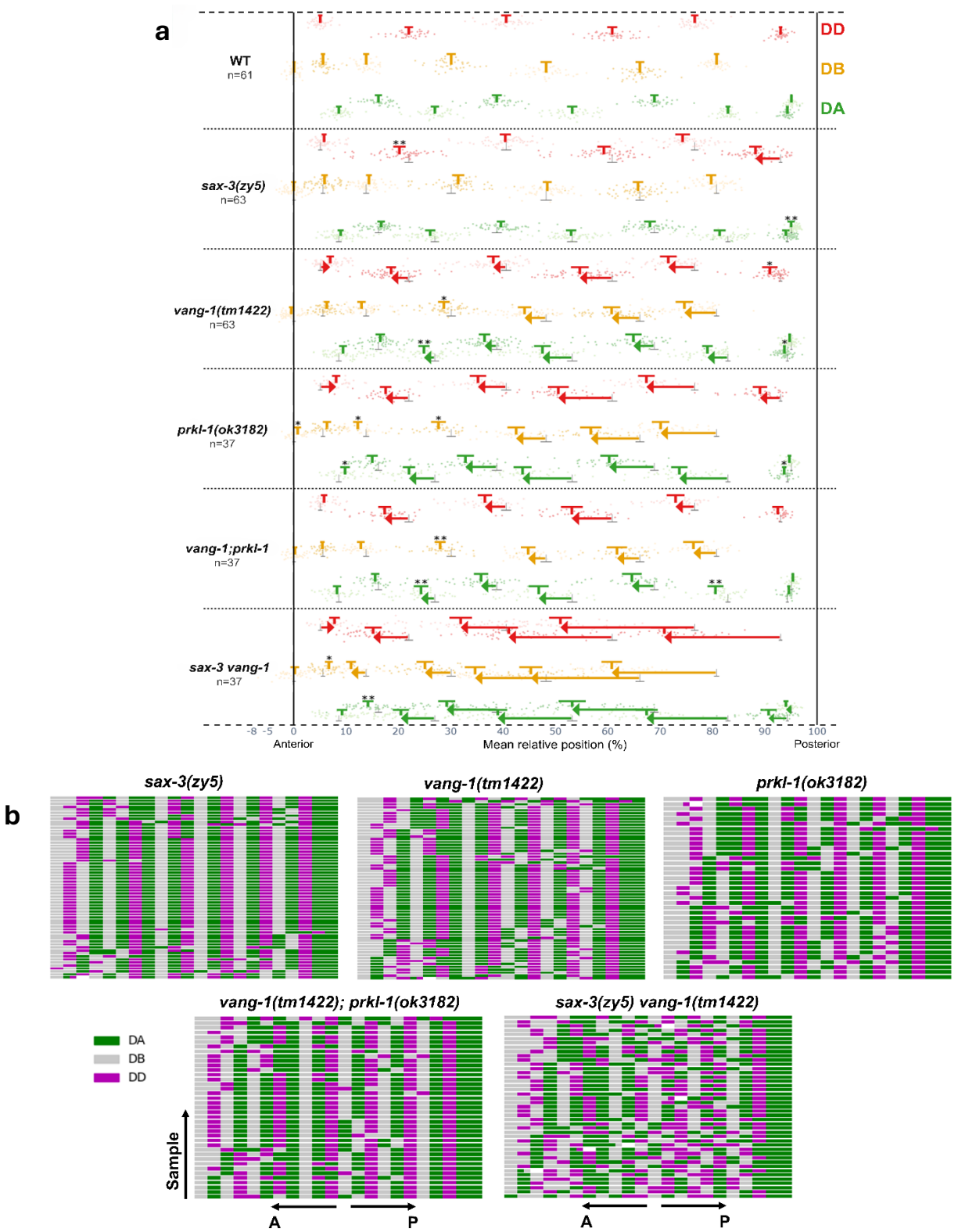


Fig. 6 | Quantification of neuron displacements in single and double mutants via VNC-Dist

(a) Mean relative distance of neuronal positions in *C. elegans* at L1 along the AP axis of VNC from SABVL (0%) to rectum (100%) are shown. Mean position of DD (red), DA (green), and DB (orange) neurons are compared to WT. Arrows indicate statistically significant, high-magnitude shifts (Cohen's d ≥ 0.7) from WT to mutant. Significance is denoted by asterisks, with unlabeled arrows representing the highest significance (***). DB1 might be anterior to SABVL, with a negative value. Error bars represent mean neuron positions ± 95% confidence intervals (CI); ***P < 0.001, **P < 0.01, *P < 0.05. **(b)** Neuron type arrangement of all 22 motoneurons from left (anterior) to right (posterior), highlighting how neuron types shift or switch across different genotypes along the AP axis. Overlapping neuron positions in mutants are indicated by skipped white bars.

Supplementary figures:

VNC-Dist: A machine learning-based pipeline for quantification of neuron positioning in the ventral nerve cord of *C. elegans*

Saber Saharkhiz[1,3,4], Mearhyn Petite[3], Tony Roenspies[3], Theodore J. Perkins[2,4*], Antonio Colavita[1,3*]

1. Department of Cellular and Molecular Medicine, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
2. Department of Biochemistry, Microbiology and Immunology, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
3. Neuroscience, Ottawa Hospital Research Institute, Ottawa, ON, Canada
4. Regenerative Medicine, Ottawa Hospital Research Institute, Ottawa, ON, Canada

***Corresponding authors**: Antonio Colavita, colavita@uOttawa.ca; Theodore J. Perkins, tperkins@ohri.ca

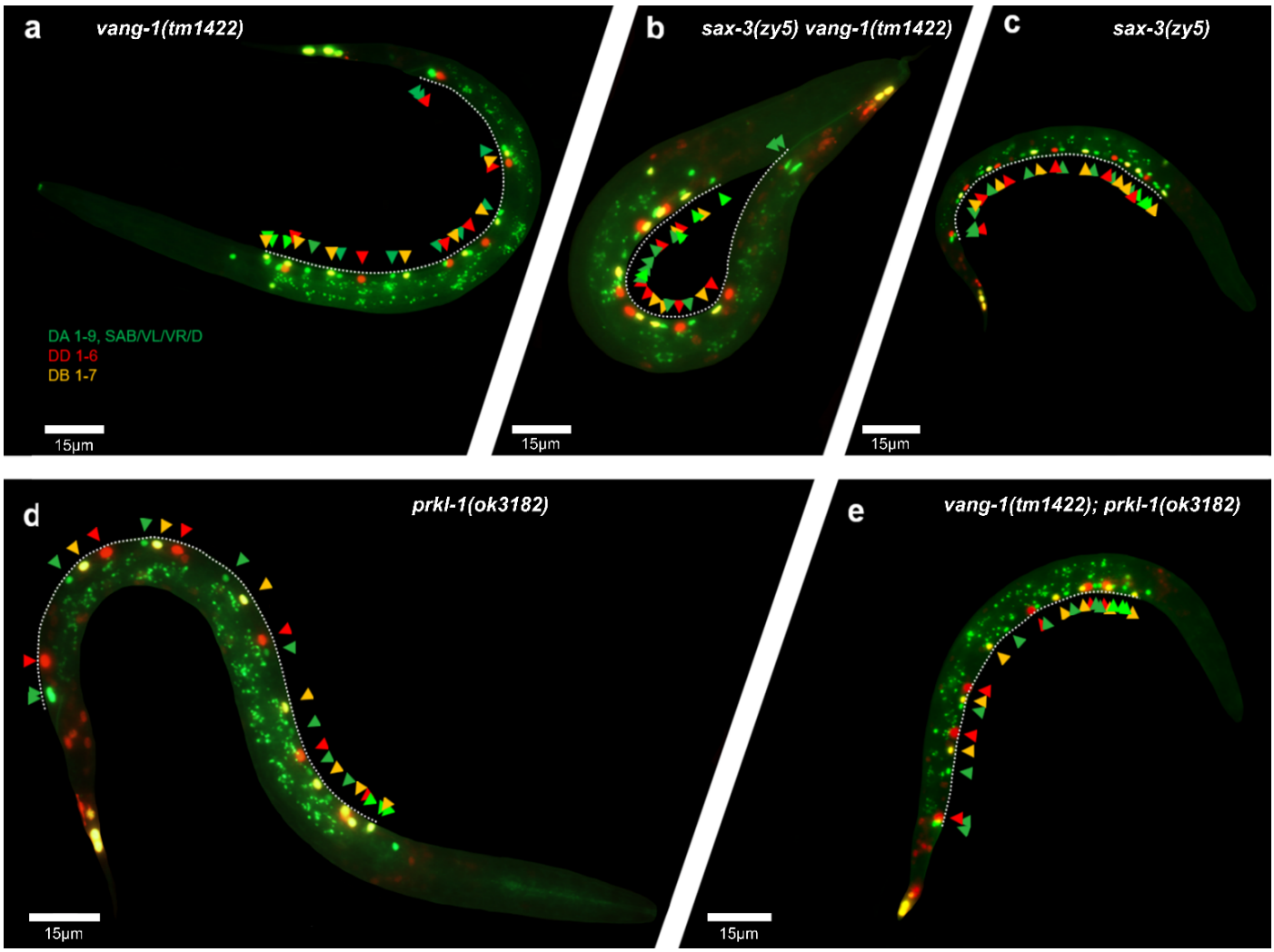


Fig. S1 | Neuronal positioning across mutant genotypes in *C. elegans*.

(a–e) Representative fluorescent images of *vang-1(tm1422)*, *sax-3(zy5) vang-1(tm1422)*, *sax-3(zy5)*, *prkl-1(ok3182)* and *vang-1(tm1422); prkl-1(ok3182)* mutants respectively. Green arrows: DAs and SABs; red arrows: DDs; orange arrows: DBs. Neuron sorting patterns along the AP axis of the VNC are not fully discernible; however, neuronal positions appear to shift and vary across genotypes. The dashed white line represents the VNCL1 curve. Note: Worms were segmented from the background.

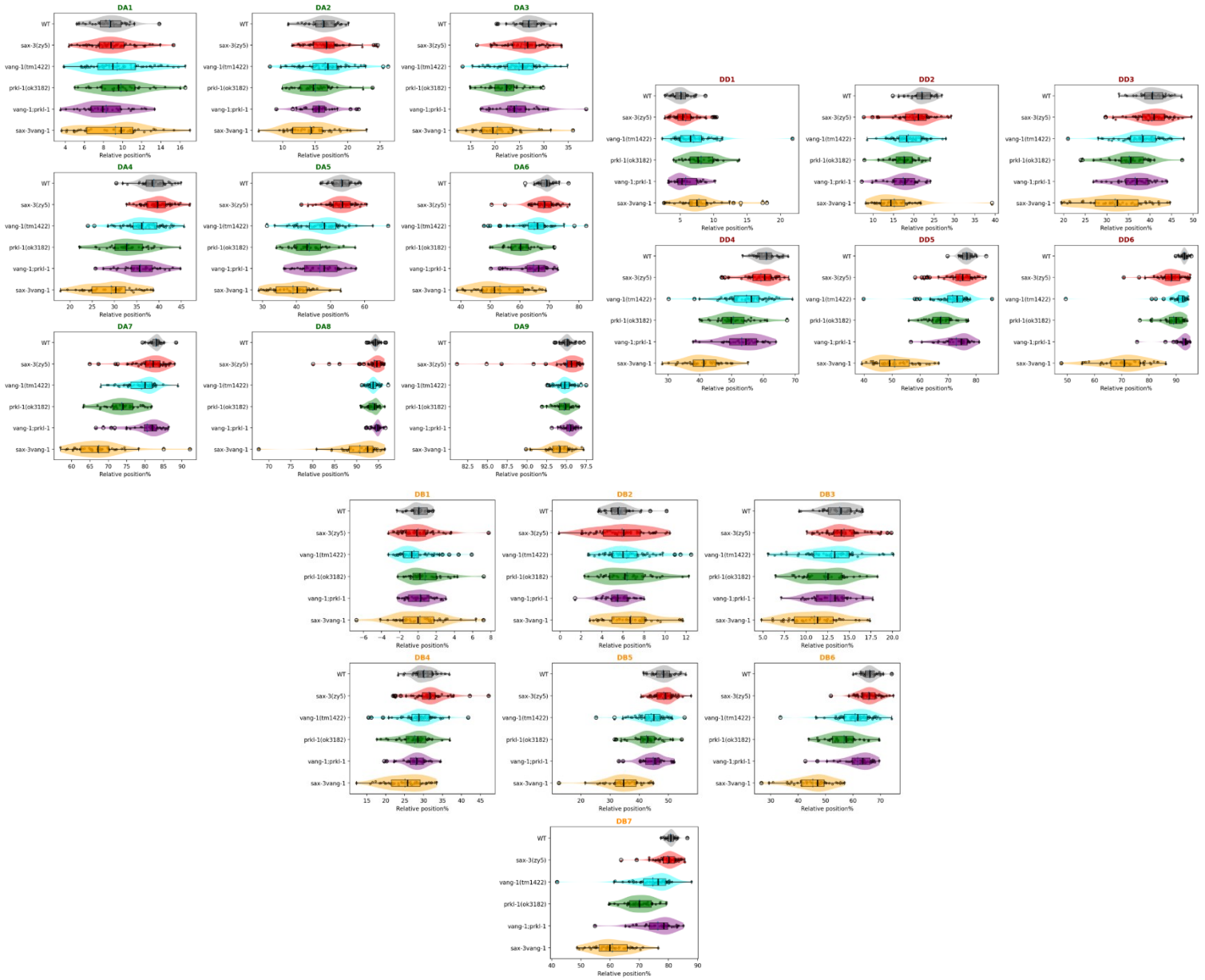


Fig. S2 | Visualization (violin plot overlaid a box plot) of relative distances (%) for DD, DA and DB motoneurons across genotypes.

Mutant strains (*prkl-1(ok3182)*, *sax-3(zy5)*, *vang-1(tm1422), vang-1;prkl-1, sax-3 vang-1*) exhibit altered anteroposterior neuron positioning compared to WT. Distributions assess normality and highlight genotype specific effects on neuron positioning along the AP axis, with median (black) and mean (blue) indicated by vertical lines.

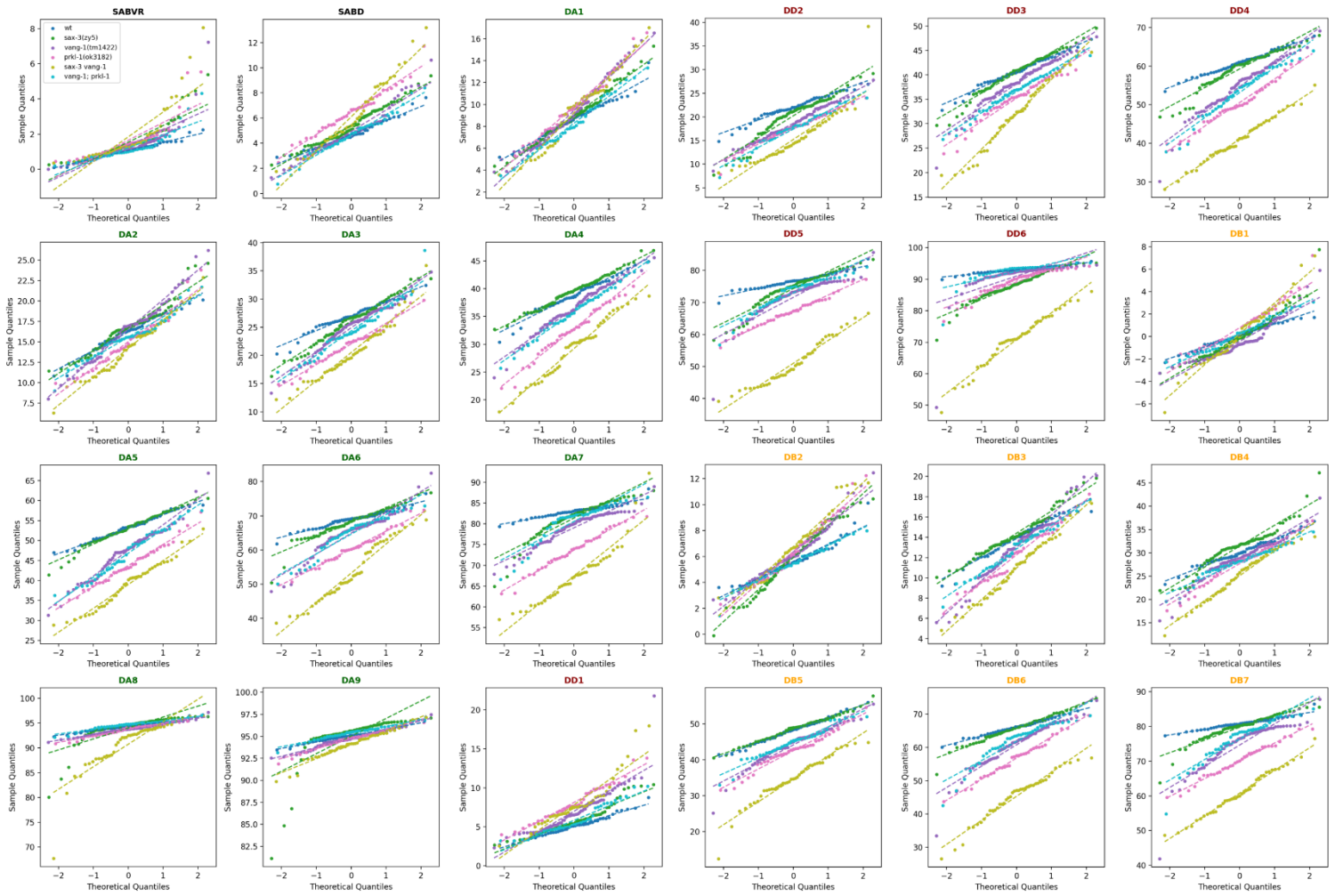


Fig. S3 | Normality assessment of relative distances (%) for DA, DD, and DB neurons in WT and mutants.

Q–Q plots are used to evaluate the normality of neuronal relative distance distributions for DA, DD, and DB neurons across WT and mutant genotypes (*prkl-1(ok3182)*, *vang-1(tm1422)*, *sax-3(zy5)*, *vang-1; prkl-1*, *sax-3 vang-1).* Deviations from the diagonal line indicate departures from normality, revealing genotype-specific effects on neuron positioning along the AP axis. Each genotype is color-coded as shown in the legend.

a

Neuron	Genotype	Shapiro_Statistic	Shapiro_P_Value	Normal
SABVR	WT	0.97895288	0.621282173	Yes
SABD	WT	0.966393962	0.24863253	Yes
DA1	WT	0.972003519	0.384630135	Yes
DA2	WT	0.96818237	0.286588291	Yes
DA3	WT	0.969255256	0.311719626	Yes
DA4	WT	0.972478111	0.398490217	Yes
DA5	WT	0.975625752	0.499636048	Yes
DA6	WT	0.968910153	0.303436853	Yes
DA7	WT	0.930145277	0.013102261	No
DA8	WT	0.961755336	0.170624775	Yes
DA9	WT	0.965732689	0.235784939	Yes
DD1	WT	0.979528717	0.643339681	Yes
DD2	WT	0.967590929	0.273503051	Yes
DD3	WT	0.984371057	0.825634641	Yes
DD4	WT	0.991779183	0.989076986	Yes
DD5	WT	0.959384296	0.140370838	Yes
DD6	WT	0.978281896	0.595847549	Yes
DB1	WT	0.964342773	0.210743129	Yes
DB2	WT	0.931954734	0.015087906	No
DB3	WT	0.956969946	0.114966676	Yes
DB4	WT	0.984945372	0.845202904	Yes
DB5	WT	0.988022379	0.933018837	Yes
DB6	WT	0.986105963	0.88207035	Yes
DB7	WT	0.953935875	0.08941936	Yes

b

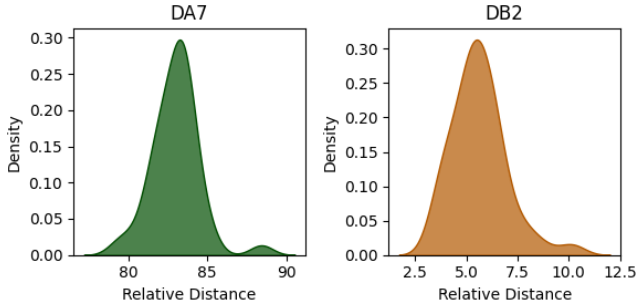


Fig. S4 | Shapiro–Wilk test results for normality of neuronal relative distances (%) in WT *C. elegans*.

(a) Neurons with p-values > 0.05 are considered normally distributed. **(b)** Although DA7 and DB2 show significant deviations from normality ($p < 0.05$), their Shapiro statistics, histograms, and Q–Q plots (Fig. S4) suggest they can be reasonably assumed to follow a normal distribution (Sample size n= 42).

Supporting information:

VNC-Dist: A machine learning-based pipeline for quantification of neuron positioning in the ventral nerve cord of *C. elegans*

Saber Saharkhiz[1,3,4], Mearhyn Petite[3], Tony Roenspies[3], Theodore J. Perkins[2,4*], Antonio Colavita[1,3*]

1. Department of Cellular and Molecular Medicine, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
2. Department of Biochemistry, Microbiology and Immunology, Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada
3. Neuroscience, Ottawa Hospital Research Institute, Ottawa, ON, Canada
4. Regenerative Medicine, Ottawa Hospital Research Institute, Ottawa, ON, Canada

***Corresponding authors**: Antonio Colavita, colavita@uOttawa.ca; Theodore J. Perkins, tperkins@ohri.ca