

Supplementary Information

The *C. elegans* Lifespan Machine

Nicholas Stroustrup¹, Bryne E. Ulmschneider, Zachary M. Nash, Isaac F. López Moyado,
Javier Apfeld^{*,1} and Walter Fontana^{*,1}

Department of Systems Biology, Harvard Medical School

* These authors are co-last authors ¹ These authors are corresponding authors
stroustr@fas.harvard.edu — javier_apfeld@hms.harvard.edu — walter@hms.harvard.edu

Table of Contents

List of Figures	2
Supplementary Note 1 — Lifespan Machine hardware	4
1.1 Image acquisition	4
1.2 Temperature control during image acquisition	5
Supplementary Note 2 — Image processing.....	7
2.1 Masking, background subtraction, and segmentation	7
2.2 Registration of consecutive scanner images	7
2.3 Worm recognition	8
Supplementary Note 3 — Assembly.....	13
3.1 Introduction	13
3.2 Complete parts list	13
3.3 Recommended scanner modifications	16
3.3.1 Overview of the fan mounting procedure	16
3.3.2 Modifying the scanner chassis	16
3.3.3 Modifying the transparency unit	17
3.3.4 Accessing and preparing the fixed scanner lens	21
3.4 Instructions for focusing a scanner	22
Supplementary Note 4 — Software toolset	25
4.1 Software components	25
4.2 The Worm Browser GUI client	25
Supplementary Note 5 — Movement analysis.....	27
5.1 Identification of stationary worms	27
5.2 Posture analysis of stationary worms	29
Supplementary Note 6 — Statistics of worm movement	31
Supplementary Note 7 — A “death phenotype”	33
Supplementary Note 8 — Censoring	36
8.1 Right censoring and interval censoring	36
8.2 Tracking population size and right censoring strategies	37
8.3 Worm clusters and interval censoring	40
8.4 Censoring strategies for worm clusters	41
8.5 Determining total population size	42
Supplementary Note 9 — Statistical power.....	43
Supplementary Note 10 — Aggregation of survival curves	45
10.1 Device-specific environmental effects on lifespan	45
10.2 The device-effect regression model	45

Supplementary Note 11 — Hazard	47
11.1 Hazard estimation and rendering	47
11.2 Hazard of wildtype	48
11.3 Hazard and sample size	51
Supplementary Note 12 — Control for scanner effect	52
Supplementary Note 13 — Additional survival curves, controls.....	54
Supplementary Note 14 — Limitations	58
Supplementary Note 15 — Resolution and scalability	60
Supplementary Note 16 — Use cases	61
Acknowledgments	62
References.....	62

List of Figures

1-1 Lifespan Machine hardware	4
1-2 The effect of scanner temperature on lifespan	5
1-3 Consistency of scanner surface temperature	5
2-1 Registration of consecutive scanner images.....	7
2-2 Feature palette for worm recognition	9
2-3 Worm and non-worm objects	10
2-4 Support Vector Machine for worm recognition	10
2-5 Image segmentation examples	11
3-1 Scanner modification diagram	18
3-2 Photographs of chassis modifications	19
3-3 Photographs of mounted scanner fan.....	20
3-4 Fan wiring diagram.....	21
3-5 Inside the scanner chassis	23
3-6 Reference images demonstrating sufficient focus	24
4-1 Worm Browser screen shot	26
5-1 Parameter robustness of posture analysis.....	30
6-1 Population statistics of movement classes	31
6-2 Stationarity (posture) span	32
7-1 The volume of a worm dips at or near death	33
7-2 Coincidence of volume dip and cessation of motion.....	34
7-3 Generality of the volume dip.....	35
8-1 Interval midpoint vs Turnbull estimator	36
8-2 Going missing and getting lost (schematic)	38
8-3 Going missing and getting lost (data).....	39
8-4 Interval censoring for worm clusters (schematic)	40
8-5 Censoring strategies for terminal worm clusters	41
9-1 Statistical power	43
11-1 Wildtype high-resolution hazard functions at 25 °C	48
11-2 Wildtype high-resolution quantile-quantile plots at 25 °C	49
11-3 Wildtype hazard at 25 °C scored manually	50
11-4 Wildtype hazard at 20 °C on HT115.....	50
11-5 Population size and hazard parameter estimates	51
12-1 Absence of a “scanner effect”	52
13-1 Wildtype at 25°, device-corrected	54

13-2 RNAi lifespans at 20 °C	54
13-3 <i>daf-16(RNAi)</i> at 25 °C	55
13-4 Automated lifespan assay of a severely uncoordinated mutant	55
13-5 Survival curves of <i>unc-50(e306)</i> and <i>unc-4(e120);rol-6(su1006)</i> at 25 °C	56
13-6 Lifespans of of temperature-sensitive self-sterile mutants	56
13-7 Visual controls of additional mutant survival curves	57
13-8 Error of LM operation at high-temperature	57

Supplementary Note 1 – Lifespan Machine hardware

1.1	Image acquisition	4
1.2	Temperature control during image acquisition	5

1.1 Image acquisition

Our reference installation consists of 50 scanners allocated to 5 incubators. A single scanner is sufficient to obtain controlled survival curves of high accuracy, while a fleet of scanners enables high throughput. In Note 16, we provide a description of use cases for varying numbers of scanners. Instructions for the modification of a commercially available scanner model (Epson v700) are provided as a separate supplementary document. At www.lifespanmachine.org, we have set up pointers to a public repository for the software components of the Lifespan Machine to facilitate user-driven development and improvement.

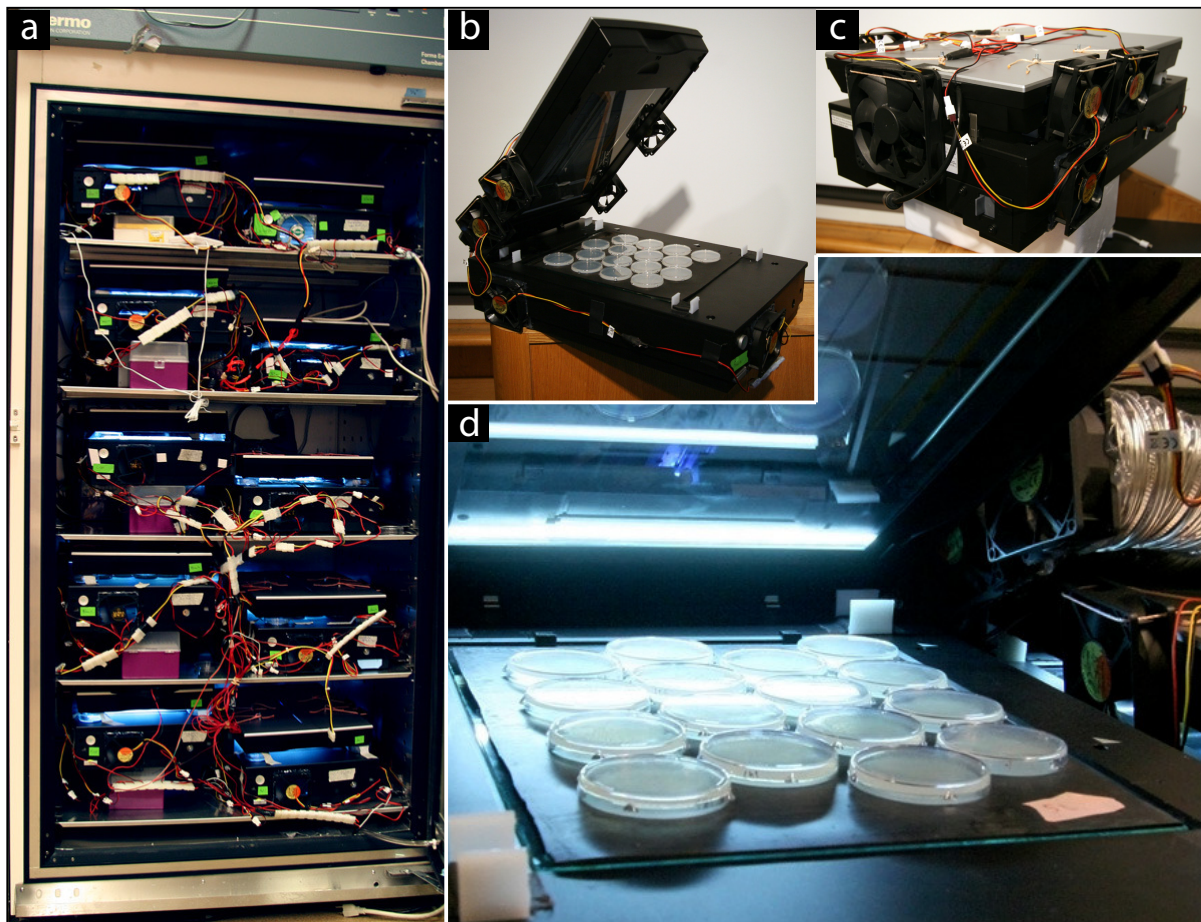


Figure 1-1: Lifespan Machine hardware. **(a)** A Thermo Forma incubator provides temperature control for ten flatbed scanners. **(b)** A scanner unit with open lid and plates loaded, showing fans mounted on the lid to create a flow of air between lid and scanner surface when the lid is closed. **(c)** A scanner unit with closed lid, as during operation, shown from rear to highlight fans mounted into the chassis. **(d)** The typical plate configuration consists of 16 plates inserted face down through holes punched into a rubber mat. The mat seals the plates to the glass surface of the scanner, stabilizing their position and preventing desiccation. Note the light source located in the lid (transillumination mode).

1.2 Temperature control during image acquisition

An important aspect of our method is the control of temperature within a tight range, reducing the large excursions that are sometimes unavoidable in the manual method where plates are moved in and out of an incubator for scoring. The notorious temperature sensitivity of *C. elegans* is highlighted by its variation between scanners (Fig. 1-2).

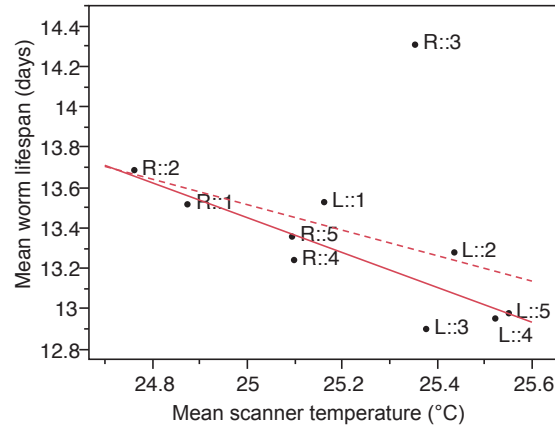


Figure 1-2: The effect of scanner temperature on lifespan. The lifespans of wildtype animals were acquired as described for Fig. 2 of the main text, and loaded onto 160 plates distributed across 10 scanners. The average lifespan of all animals on a given scanner was compared to the temperature measured on that scanner (Online Methods). A linear regression was run with all scanners ($r^2 = 0.163$, red dashed line) and with the R::3 outlier excluded ($r^2 = 0.746$, red solid line).

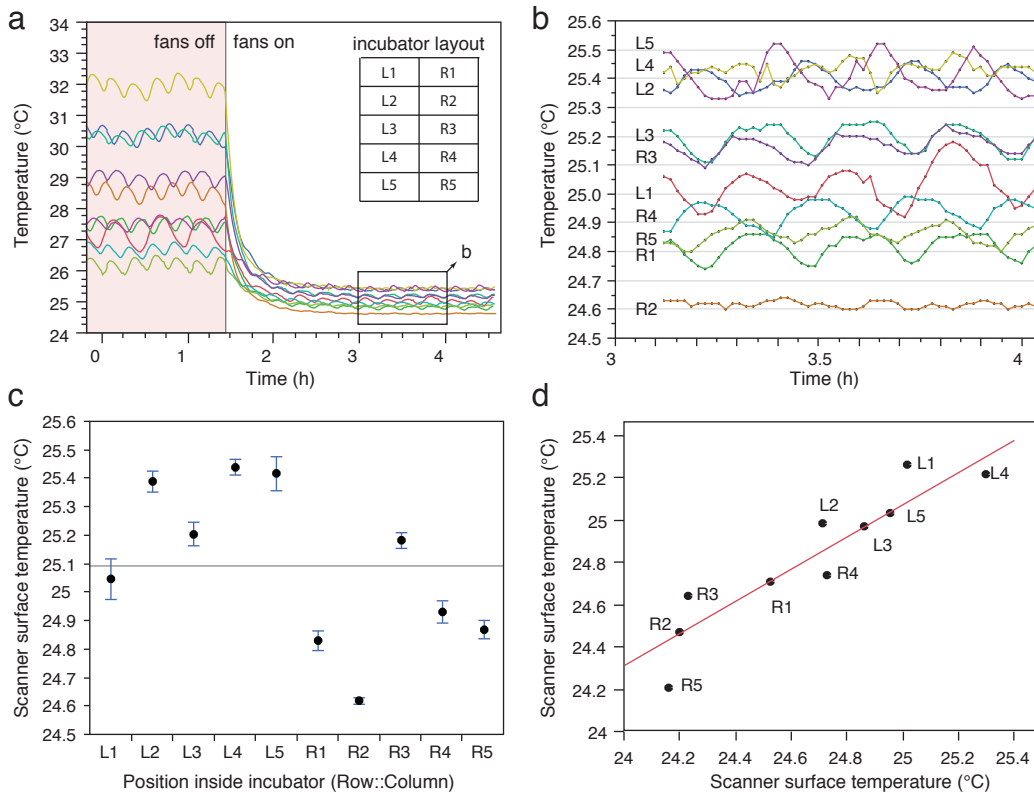


Figure 1-3: Consistency of scanner surface temperature. Caption on next page.

Figure 1-3 (Cont'd): (a) Local temperature differences between and temperature excursions of 10 scanners operating at a scan frequency of 15 minutes inside an incubator (Online Methods). On the left, the fans are turned off; on the right, they are turned on. The inset diagrams the shelving of scanners for reference in panels (b)-(d). **(b)** Detail of the rectangular area indicated in panel (a). Fans reduce local temperature differences due to incubator airflow to within 1°C and temperature excursions from scanning to within 0.2°C. **(c)** Data shown in panel (b) are reported against the location of the scanners in an incubator adjusted to maintain 25°C, with 5 shelves and 2 scanners (Left and Right) per shelf; see inset of panel (a). Time series are summarized by the mean with error bars indicating one standard deviation. The left side of the incubator has a slightly higher average temperature compared to the right side due to incubator airflow. **(d)** In a separate experiment, the scanner temperatures of a single incubator were measured at one time (x-axis) and again two months later (y-axis). The temperature measures correlate ($R^2 = 0.85$), indicating stability of operating conditions.

Temperature control is especially important when measuring lifespans outside the “room temperature” range, as with thermotolerance assays (e.g. Fig. 5 of the main text). To control temperature gradients that result from a scanner’s electronics and light source, cooling fans are mounted into the chassis of the scanner, as shown in Fig. 1-1b and schematically in Fig. 1 of the main text. Fig. 1-3 below shows the efficacy of these modifications.

Supplementary Note 2 — Image processing

2.1	Masking, background subtraction, and segmentation	7
2.2	Registration of consecutive scanner images	7
2.3	Worm recognition	8

2.1 Masking, background subtraction, and segmentation

In images captured by a scanner, only areas corresponding to the agar surface of each plate need to be analyzed. “Masking” is the process of separating multiple plates and removing their edges. The Worm Browser (Note 4 section 4.2 and Supplementary Video 6) generates a low-resolution composite of each image captured. The user can then draw on top of this image to produce an overlay (the mask) that specifies the location of each plate. This drawing can be done quickly in programs such as Photoshop or GIMP [1]. (This task is a candidate for future automation.) The image-processing server uses the resulting mask to extract individual plate images and stores them in a stereotyped folder hierarchy associated with each experiment.

Prior to analysis, all images $X^{(t)}$, $t = 1, \dots, T$, undergo background subtraction. Each pixel $x_i^{(t)} \in X^{(t)}$ is reduced by the median value in a surrounding square of 25×25 pixels $198 \mu\text{m} \times 198 \mu\text{m}$. This median filtering removes variation at spatial scales larger than roughly twice the average width of an adult worm ($198 \mu\text{m}$) suppressing background effects including shadows, smooth bacterial lawns, and uneven agar shading.

Median filtering sets the stage for image segmentation—the separation of worms from their agar and bacterial surface background—using a global threshold. Two global thresholds are applied to images, one restrictive threshold that identifies only dark objects such as worms, but tends to break worms into multiple objects, and a second more permissive threshold that tends to pick up extraneous features of the bacterial lawn. The two are compared to produce a final image segmentation. See also Fig. 2-5.

2.2 Registration of consecutive scanner images

The panels in Fig. 2-1 show the magnitude of two types of jitter in the acquisition of images by our scanners (and presumably consumer electronics scanners in general). Displacements like these must be taken into account when processing images leading up to the movement analysis (sections 5.1 and 5.2).

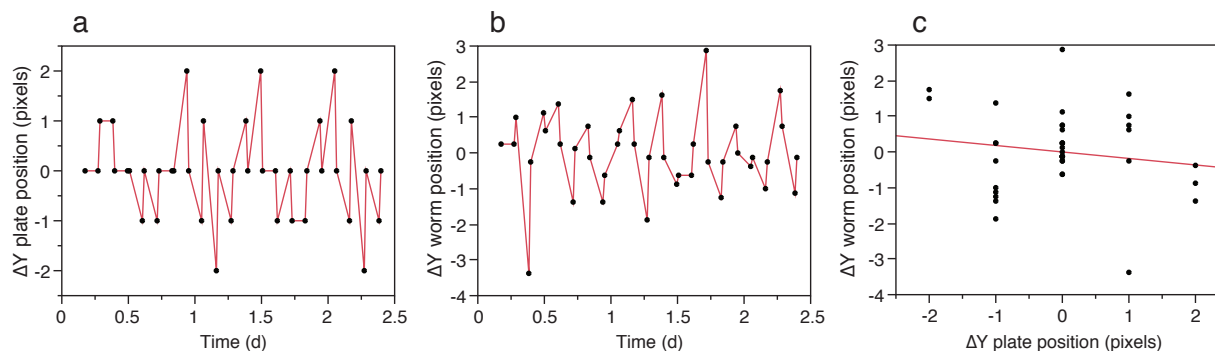


Figure 2-1: Registration of consecutive scanner images. **(a):** The mechanism advancing the scanning bar causes offsets of a few pixels relative to world coordinates between subsequent images of the same plate. **(b):** Slight distortions in plate images cause displacements of a few pixels relative to plate coordinates for stationary worms. **(c):** The two types of displacements shown in panels (a) and (b) are uncorrelated.

2.3 Worm recognition

The image-processing pipeline computes for each foreground object a set of features that are subsequently used by a Support Vector Machine (SVM) [2], Fig. 2-4, to discriminate between worm and non-worm objects, Fig. 2-3. The following is a list of 65 features used by SVM-based worm classifier. Their distribution of values is shown in Fig. 2-2.

- Pixel Area
- Rectangular Diagonal
- Max Width
- Width at Center
- Spine (SP) Length / Pixel Area
- SP Length / Rect Diagonal
- SP Length / Width at Front
- Average Curvature
- Curvature X Intercepts
- Absolute Intensity (AI)
- AI Roughness (Entropy)
- AI Average Along SP
- AI darkest 20% area
- AI of Region
- AI Normalized Avg Along SP
- RI Skew
- RI Max
- RI darkest 20% of pixels
- RI Variance Along Spine
- RI Normalized Max
- Edge Area / Object Area
- Intensity Profile Max
- Rectangular Width
- Spine Length
- Min Width
- Width at Front
- SP Length / Rect Width
- SP Length / Max Width
- SP Length / Width at Rear
- Max Curvature
- Curvature Variance
- AI Variance
- AI Roughness
- AI Average in Nbrhd
- AI Dist from Nbrhd
- AI Normalized Average
- Relative Intensity (RI)
- RI Roughness (Entropy)
- RI Average Along SP
- RI darkest 20% area
- RI of Region
- RI Normalized Avg Along SP
- Intensity Profile at Edge
- Intensity Profile Variance
- Rectangular Height
- Average Width
- Variance in Width
- Width at Rear
- SP Length / Rect Height
- SP Length / Average Width
- Ratio of end widths
- Total Curvature
- Distance between ends
- AI Skew
- AI Maximum
- AI darkest 20% of pixels
- AI Variance Along SP
- AI Normalized Max
- RI Variance
- RI Roughness
- RI Average in Nbrhd
- RI Dist from Nbrhd
- RI Normalized Average
- Edge Area
- Intensity Profile at Center

Because no single feature is sufficient to accurately discriminate between worm and non-worm objects, information from multiple features must be integrated. To assess whether accurate discrimination is feasible, we hand-annotated $\sim 4,000$ images as depicting either a worm or a non-worm object. We divided this set into two parts, a training set (1489 worms, 945 non-worms) and a test set (992 worms, 631 non-worms). Using the `libsvm` library, we built an SVM model of the training data, and evaluated its performance on the test set. The confusion matrix is shown below, demonstrating a discrimination accuracy of 97.7%. The Worm Browser (Note 4 section 4.2) allows rapid inspection of all objects predicted by the machine to be worms. Mistakes can therefore be annotated, producing a final false positive rate of nearly zero.

		SVM prediction	
		worm	non-worm
by-hand	worm	991	1
	non-worm	36	595

To visualize the feasibility of discrimination, we computed the eigenvectors of the feature correlation matrix of the gold standard and represented each animal as a dot projected onto the subspace spanned by the eigenvectors with the three largest eigenvalues (principal components PCA 1-3), as shown in Fig. 2-4.

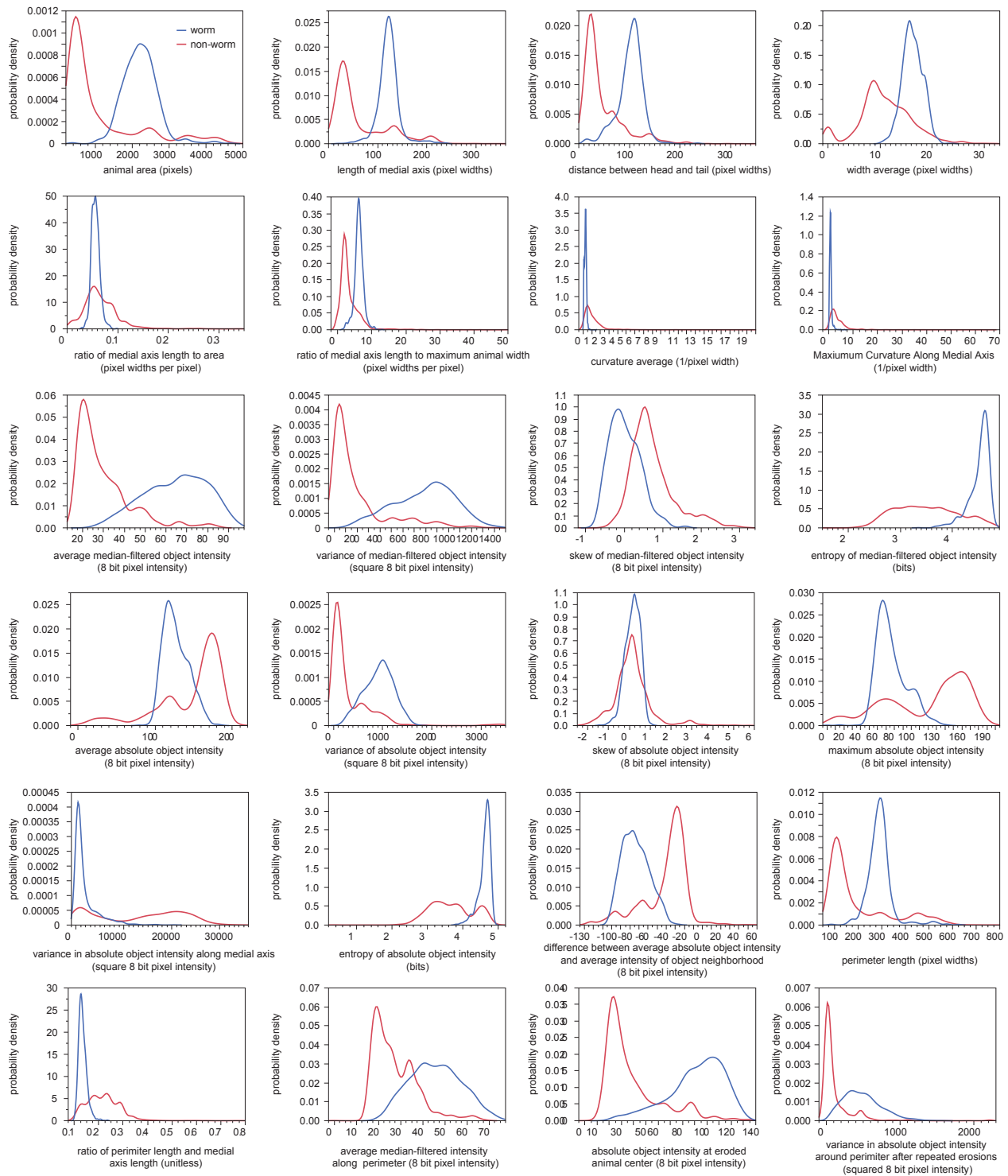


Figure 2-2: Feature palette for worm recognition. The panels show the frequency distribution of the values for each feature in the case of worms (blue) and non-worms (red). Because feature value distributions of worm and non-worm objects overlap in all cases, no single feature is sufficient for accurate discrimination. 1 pixel width corresponds to $8\mu\text{m}$.



Figure 2-3: Worm and non-worm objects. The images inside the red rectangle exemplify worm objects; the images on the left, outside the rectangle, illustrate various non-worm objects ranging from fibers, to dust, to lot numbers stamped on plates. The two classes of objects are well-separated by the Support Vector Machine, Fig. 2-4, based on the feature set shown in Fig. 2-2.

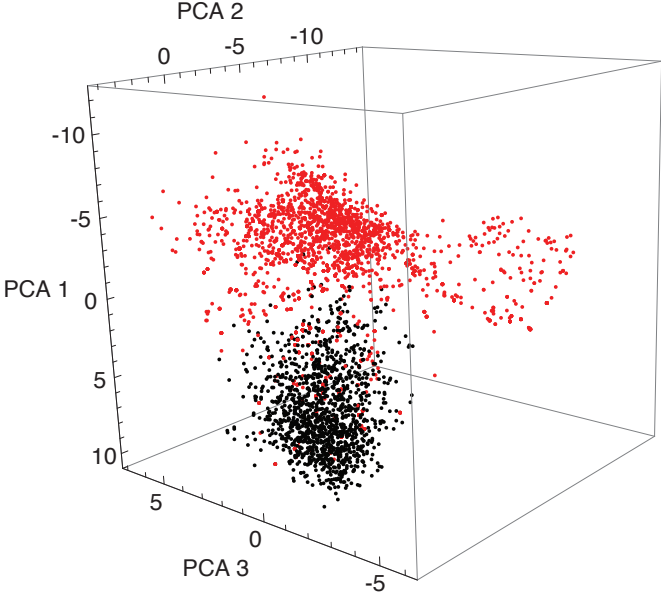


Figure 2-4: Support Vector Machine for worm recognition. Black: worms, Red: Non-worms. Each object of a hand-annotated “gold standard” is represented as a dot in a 65-dimensional feature space. The figure shows the projection of this point cloud onto the subspace spanned by the three principal components that explain the largest share of total variation (60.4%).

The previous table was based solely on wildtype. The following table illustrates the performance of the SVM when trained on a mix of strains and then applied to each separately (rather than trained on each specifically). For this analysis, we (1) lumped a number of genotypes into one big set; (2) assigned randomly 3/5 of animals to the training set and the remainder to the test set; (3) trained an SVM on the training set; (4) tested it on the test set; (5) calculated the confusion matrix for each

genotype. FP=false positives, FN=false negatives:

	all	<i>daf-16 (mu86)</i>	wildtype	<i>glp-1</i>	<i>age-1(hx546)</i>
FP	84/1550 = 5.4%	23/207 = 11.1%	21/562 = 3.7%	32/453 = 7.1%	8/328 = 2.4%
FN	73/2426 = 3.0%	11/336 = 3.3%	19/777 = 2.4%	31/883 = 3.5%	12/430 = 2.8%

All strains have a false negative rate between 2.4% and 3.5%. The false positive rate is more varied, but ultimately immaterial, since we catch the non-worm objects during visual inspection in the quality control phase with the Worm Browser.

Fig. 2-5 provides examples of image segmentation and object recognition using the SVM, illustrating the difficulties of disentangling worm clusters. Refer to sections 8.2 and 8.3, as well as Figs. 8-3 and 8-5, for more detail on the accounting of multi-worm clusters. Supplementary Videos 1–5 provide time-lapse movies of plates with wildtype, *age-1(hx546)*, and *unc-64(e246)* mutants for the entire duration of an experiment, both with and without metadata overlay.

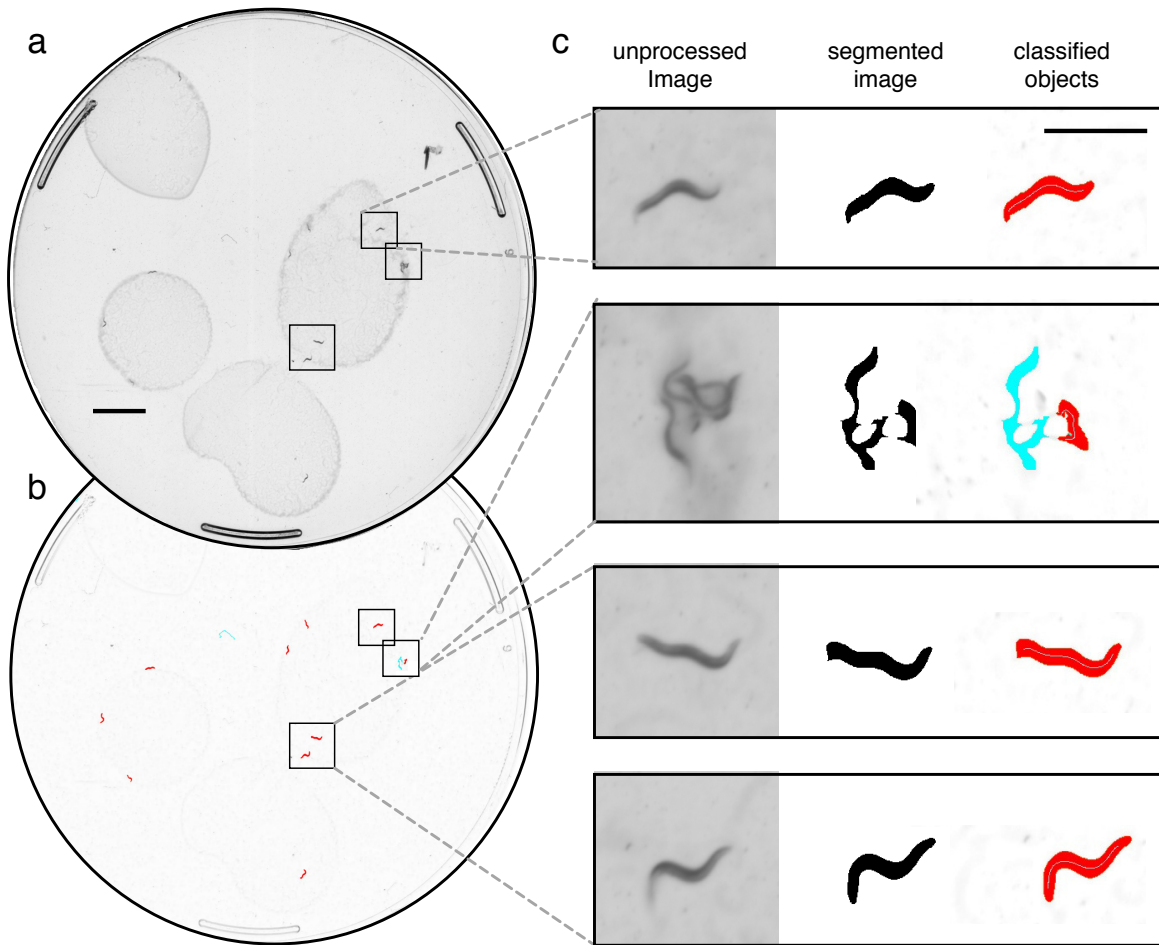


Figure 2-5: Image segmentation examples. Caption on next page.

Figure 2-5 (Cont'd): (a) A scanner raw image of an agar plate with worms on patches of bacterial lawn (*E. coli* OP50). Scale bar on the left corresponds to 5mm; the scale bar on the right to 1mm. **(b)** The same image after background subtraction (using a median filter), with foreground objects resulting from image segmentation highlighted in color. Objects classified as worms by the SVM are colored red; non-worms are colored blue. **(c)** Enlarged images of worms indicated in panel (b), as they appear in consecutive image processing steps. The scale bar marks 1mm. The first, third, and fourth rows are examples of correctly processed animals. The second row illustrates the difficulties of parsing multiple worm clusters. Note how several of the worms are incorrectly identified. Such errors, when they occur in the context of dead worms, can be rapidly identified in the Worm Browser (Supplementary Video 6).

Supplementary Note 3 — Assembly

3.1	Introduction	13
3.2	Complete parts list	13
3.3	Recommended scanner modifications	16
3.3.1	Overview of the fan mounting procedure	16
3.3.2	Modifying the scanner chassis	16
3.3.3	Modifying the transparency unit	17
3.3.4	Accessing and preparing the fixed scanner lens	21
3.4	Instructions for focusing a scanner	22

3.1 Introduction

The lifespan machine identifies worm death times through the combination of four interacting components:

- (i) An image acquisition system based on modified flatbed document scanners;
- (ii) An image acquisition control server running under Linux on a PC;
- (iii) An image analysis server that can run in the background of any x86 Windows, Apple, or Linux computer;
- (iv) A data annotation and aggregation client program that allows rapid inspection and annotation of automated results.

This document contains specifications for a reference implementation of the image acquisition system. An up-to-date, detailed specification of the software is provided as part of the source code documentation.

These instructions are lengthy, representing the authors' attempt at assembling a comprehensive, self-sufficient document. The modifications themselves are actually quite straightforward. A research assistant (RA) with a life science undergraduate degree and no engineering background learned how to perform all modifications over the course of several hour-long training sessions. The modifications required to complete two scanners, capable of monitoring over a thousand individuals, should take a novice two or three half days. All modifications are easily performed on a standard lab bench.

The authors do not currently possess good estimates for the operating lifespan of a modified scanner. Existing devices appear to operate correctly after nearly two years of continuous use. Several have been in operation for nearly five years. Across a fifty-scanner installation, the authors have retired only three scanners as a result of malfunction, usually after only a few weeks of operation. Presumably, scanner fluorescent bulbs will eventually require replacement.

3.2 Complete parts list

The following list details the equipment required for a reference, 10 scanner installation capable of monitoring 5600 nematodes simultaneously. Smaller or larger clusters can be easily assembled at proportional cost. All prices are from circa 2012 and approximate.

10x Epson v700 Perfection V700 Photo Scanners.

Available from Amazon.com or Epson.com for \$540. In the past, we have also used the discontinued Epson 4990 Photo scanner. Other scanner models that possess appropriate transillumination capabilities are likely to be adaptable.

70x 80x80x25mm DC fans

This is a standardized computer chassis component, and as such is widely available from

many suppliers. For example, the Thermaltake DuraMax 8 AF0058 80mm Case Fan available from NewEgg.com for \$8.20.

10x 120x120x25mm DC fans

This is a standardized computer chassis component, and as such is widely available from many suppliers. For example, the Thermaltake A2492 120mm Case Fan is available from NewEgg.com for \$9.

1x A 300+Watt source of 12V DC Power

We use standard Power Supply Units (PSU), which are mass-produced components of PC computer chassis, and as such are widely available at low prices from many suppliers. For example, the CORSAIR Builder Series CX430 V2 (CMPSU-430CXV2) 430W is available from NewEgg.com for \$45

1x Temperature-controlled environment for equipment

This might be an incubator or a temperature-controlled room. For example, the Thermo Forma Environmental Chamber is a multi-use incubator which many labs already possess. This incubator, in 2010, cost approximately \$13,000 new, and as such represents more than half the price of an incubator-based Lifespan Machine installation.

50x Machine Screws

For example, Stainless Steel Machine Screw and Hex Nuts 11/32x1/8" 8-32 available from McMaster-Carr; a 100-unit box for \$4.25

50x Machine Screw Nuts

For example, Stainless Steel Machine Screw and Hex Nuts 11/32x1/8" 8-32 available from McMaster-Carr; a 100-unit box for \$4.25

80x Nylon twine or string, cut into 16.5" lengths

Used for mounting fans. Simpson's Individual Stringettes; available from McMaster-Carr; 1800 feet for \$8.00

1x Reasonably modern (circa 2012) x86 PC or laptop capable of running Linux

More details provided under "Software Components" (section 1.3.1 of the scientific Supplementary Information); \$650

10x 9x11.5" inch glass sheets

Standard window pane glass, available from most local hardware stores; \$5 per sheet.

10x 8.7x11.3" inch rubber mats

Cut from rubber stock, for example High-Strength EPDM Rubber Sheet, 40A, Black, 36"x36"x1/16" available from McMaster-Carr at \$20 per foot.

40x Plastic spacers to support the scanner lid

1" segments cut from 8' Polyethylene U-Shaped Channels. Leg Length - 3/4", Base Length - 5/4". Available for McMaster-Carr; \$18.8 for eight feet.

2x Gaffer's tape

This is similar to duct tape, but better in every way. Available from McMaster-Carr; \$10 a roll. Black looks sleek; pink looks totally rad.

10x Wire to power fans

20 AWG gauge or lower.

10x Rubber mats

Plates should be sealed to prevent desiccation. Many methods for doing this are possible, we use a design cut into High-Strength EPDM Rubber Sheet, 40A, Black, available from McMaster-Carr at \$22 per 3 feet. A reference design, appropriate for laser-cutting, is provided.

1x Round Hole Arch Punch 2" Diameter —*Optional tool for construction*

If a laser-cut design is not available, rubber mats can be manufactured by hand using this punch. Available from McMaster-Carr for \$70.

160x Small Petri Dishes —*Consumed by each 5600 worm experiment*

BD Falcon Petri Dishes, 50x9mm, Tight-Fit Lid, Sterile, BD Biosciences. Available from multiple suppliers; \$215 per box of 500.

40x Butt-splices or wire nuts —*Optional*

To avoid soldering joints, butt-splices or wire nuts can be used to splice fan wiring. Available from McMaster-Carr in large packs for a few dollars. Yes, these have a silly name.

1x Mini screw driver set for scanner disassembly —*Tool for Construction*

Already present in most labs. For example, a 5-Piece Phillips Screwdriver Set available from McMaster-Carr for \$22.18.

1x Spiral saw for cutting holes in scanner chassis —*Tool for construction*

Other types of cutting devices may work well, but a spiral saw cuts quickly and will impress your friends. For example, Rotozip Spiral Saw Electric, 5.5 Amp, 15000 - 30000 rpm, 9-1/4" L, available from McMaster-Carr for \$106.25.

1x Hot-Melt Glue Gun —*Tool for construction*

Standard Duty Hot-Melt Glue Gun available from McMaster-Carr for \$23. Packs of Hot melt glue sticks available from McMaster-Carr for \$10.

1x Soldering Iron or Gun —*Optional tool for construction*

For example, a Soldering Gun available from McMaster-Carr for \$58

1x A power drill / screw driver —*Optional tool for construction*

Unless you have wrists of steel, it is easier to use a power drill to add and remove machine screws. Available from McMaster-Carr for \$150, or for less at a home improvement store like Lowes.

1x Canned air —*Optional tool for construction*

A source of compressed air, either from a can (often sold to clean computer keyboards, sold at any office supply store), or from a lab air outlet.

10x Additional 80x 80x 25mm DC fans —*Optional*

Used to circulate air throughout incubator to achieve best-possible temperature calibration. Only required for certain experimental designs.

10x Metal ductwork —*Optional*

Used to circulate air throughout incubator to achieve best-possible temperature calibration. Only required for certain experimental designs. For example, 3" Aluminum Bend-and-Stay Duct Hose, available from McMaster-Carr for \$5.25 Each.

2x Plastic bin for precise drying of plates —*Useful ancillary equipment*

Exact dimensions are not crucial, but long and flat boxes work best. For example, the 35-5/8" x 18-1/4" x 6-1/4" h Long Underbed Box with Wheels, available from The Constainer Store . com for \$20.

- 2x Sheet of Wire Mesh, fitted to plastic bin (i.e. 18"x36") —*Useful ancillary equipment*
Mounted in plastic bins for precise drying of plates. Available at local hardware stores. Corrosion-Resistant Mesh Fencing is available from McMaster-Carr for \$4.70 per square foot.
- 1x 25lbs Regular (non-indicating) 8-mesh Drierite —*Useful ancillary equipment*
Available from drierite.com for \$83.38.

3.3 Recommended scanner modifications

In this section we present the steps required for modifying stock Epson v700 Perfection Photo Scanners into a device capable of producing data of the quality presented in our manuscript. The modifications accomplish two goals: (1) install fans as a means for regulating scanner surface temperatures and (2) raise the focal plane of the scanner to optimize its use in scanning agar plates.

We see great potential for the future development of simpler, more effective strategies to accomplish these goals. The authors consider the current design primarily as a starting point from which the technology can be adapted according to each individual researcher's needs.

3.3.1 Overview of the fan mounting procedure

The Perfection v700 Photo Scanner has two parts—a bottom chassis, which contains the CCD image sensor, and a detachable lid (also called the Transparency Unit or “TPU”), which contains the transilluminating light source. To prevent the build-up of heat within the chassis, we mount three 8cm computer fans into the sides of the bottom chassis (labelled F1, F2, and F3 in the figures). These serve to move the heat produced during each scan out of the scanner body. To prevent the build-up of heat within the TPU, we hang four fans (F4, F5, F6, and F7), which serve both to remove heat from inside the TPU and to circulate cold air across the plates. A single 12cm fan (F8) is hung from the rear of the TPU, which serves to remove heat from the scanner TPU, the scanner chassis, and moves cold air across the plates.

Four cuts are made into the scanner chassis, and three into the TPU, to permit cold air to cycle through both compartments. These cuts are easily made using a spiral saw or other cutting device. Our experience is that most Dremel tools aren't quite up to the task of cutting through the hard chassis plastic. When working with any cutting tool, wear safety goggles and always pay attention. Clean, uncluttered work spaces promote safe practices.

Three fans—F1, F2, F3—are sealed in place using a glue gun. Four fans are hung from the scanner TPU are suspended by a loop of nylon twine, hanging from machine screws drilled into the TPU at positions S1, S2, S3 and S4. The holes can be cut with a power drill if one is available, or else using the spiral saw.

The exact location and dimension of each modification are presented in Figure 3-1. Photographs of the cut holes and glued fans are provided in several consecutive figures.

3.3.2 Modifying the scanner chassis

- (1) Connect an assembled scanner to a computer and scan a piece of paper, to confirm that the hardware functions. Scan in “transparency” mode to confirm that the light source in the TPU functions properly.
- (2) Disconnect the scanner and remove the TPU.
- (3) The top surface of the scanner chassis is secured by several screws, located under small plastic covers in each corner. Pop out the plastic caps and release the screws. Pop off the top surface

of the scanner and place it somewhere safe. Be careful not to get finger prints on the bottom of the scanner glass.

- (4) Using your hand, gently push the scanner bar 3/4 of the way toward the front of the scanner, to move it out of harm's way.
- (5) Place a piece of tape over the long slit opening of the scanner bar, to prevent plastic chips from entering the bar.
- (6) Using a spiral saw or your favorite cutting device, cut holes H1, H2, H3 and H4. Cuts H1 and H3 run somewhat close to internal scanner wiring. The cuts shouldn't come extremely close, but be careful and fold any nearby cables out of harm's way. Any time you use a spiral saw, wear safety goggles and stay aware. The spiral saw tends to throw little chips of plastic.
- (7) Take an 8cm fan, and use it to confirm that a fan will fit snug into each hole. Enlarge the holes where necessary. However, don't remove more plastic than required, as doing so will make it harder to hot-glue the fans in place.
- (8) Using compressed air, either from a can or from a hose hooked up to house air, blow away all the plastic chips produced by the cuts. If ignored, these plastic chips have the nasty habit of working their way into moving parts, in particular the scanner's power button. If this button ever stops working, there's probably a chip in it somewhere.
- (9) Place a 8cm fan into hole H1. Make sure the power cable is facing towards the front of the scanner, preferably from the upper right hand corner of the fan. Fan F1 should face outwards—the airflow should be heading into of the scanner. Glue it in place along all four sides. You may need to hold the fan in place with one hand and glue with another. In the same way, glue fans F2 and F3 into holes H2 and H3. Fan F2 and F3 should both be facing outwards.
- (10) Remove the protective tape added earlier to the scanner bar.

3.3.3 Modifying the transparency unit

- (1) Put the scanner chassis aside, and consider the scanner TPU. The bottom of the TPU is secured to its the cop cover by eight screws. Release them and pop off the bottom cover, and place it somewhere safe
- (2) Make cuts H5–H7. Note that the TPU connector is located at the center of H7. Remove this cable from its notch, and make cut H7. You might consider cutting a new notch somewhere next to cut H7, so that this cable can be again secured.
- (3) Using a power drill or the spiral saw, drill holes S1 through S5. Make sure to use a bit appropriate for the machine screws you are using, as you'll want the screws to fit snugly.
- (4) Using compressed air, either from a can or from a hose hooked up to house air, blow away all the plastic chips produced by the cuts.
- (5) Insert machine screws into each hole S1–S5. The head of the screw should be inside the TPU. Secure the screw by tightening a nut on the reverse side (the outside top of the TPU).
- (6) Grab the scanner TPU bottom, and snap it back into place. Screw in all screws.
- (7) Cut five 16.5" inch segments of nylon twine. Put the TPU of a scanner on a flat surface, with the machine screws facing upward. Thread one end of a piece of thread through the two holes at the top of a 8cm fan. Tie a knot at the appropriate piece of the thread such that the fan hangs snug at the TPU's side when hung from machine screw S1. Repeat three more times, hanging fans from machine screws S2–S4. Remember to orient the fans correctly; Fans F4 and F5 face inward; fans F6 and F7 face outward.
- (8) In the same way, tie a 12cm fan F8 to machine screw S5. It should face inward.
- (9) Finished!

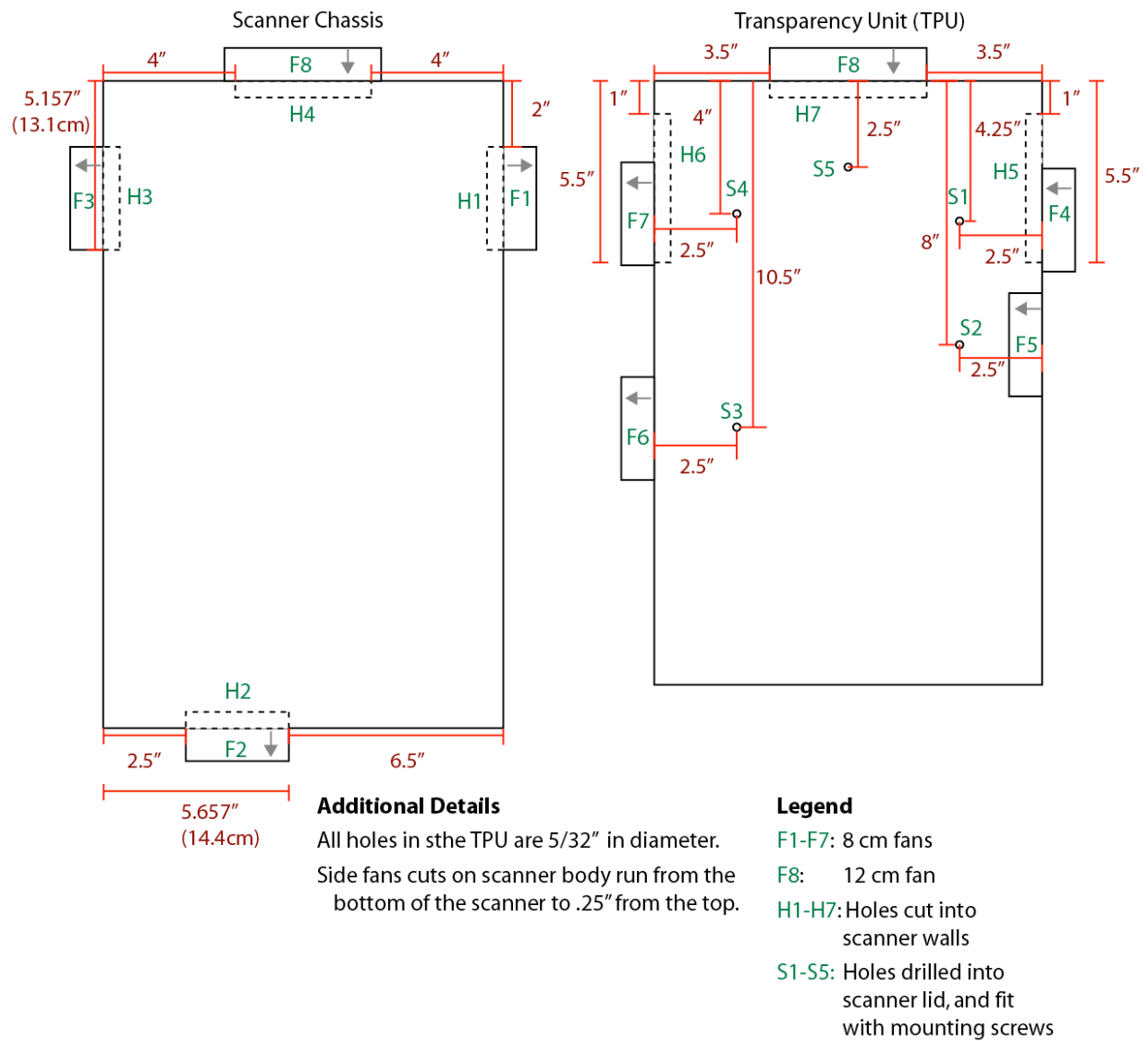


Figure 3-1: Scanner modification diagram. The location and dimensions for each cut are illustrated, for both the scanner chassis and the Transparency Unit (TPU).

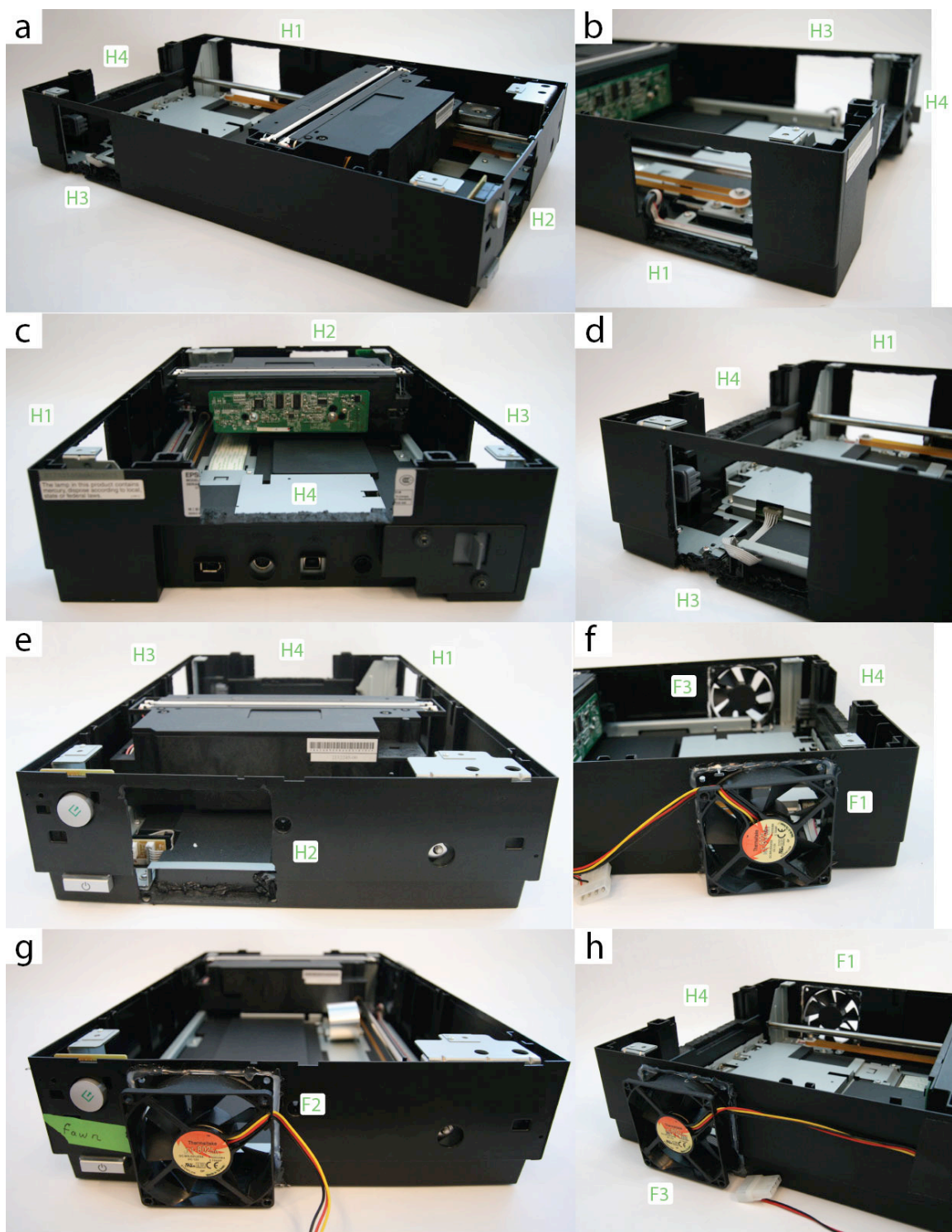


Figure 3-2: Photographs of chassis modifications. **(a):** The scanner chassis with its lid removed, and four cuts, H1–H4, made into its walls. **(b–e):** Close up views of each cut. **(f–h):** 8cm fans are glued into holes H1–H3.

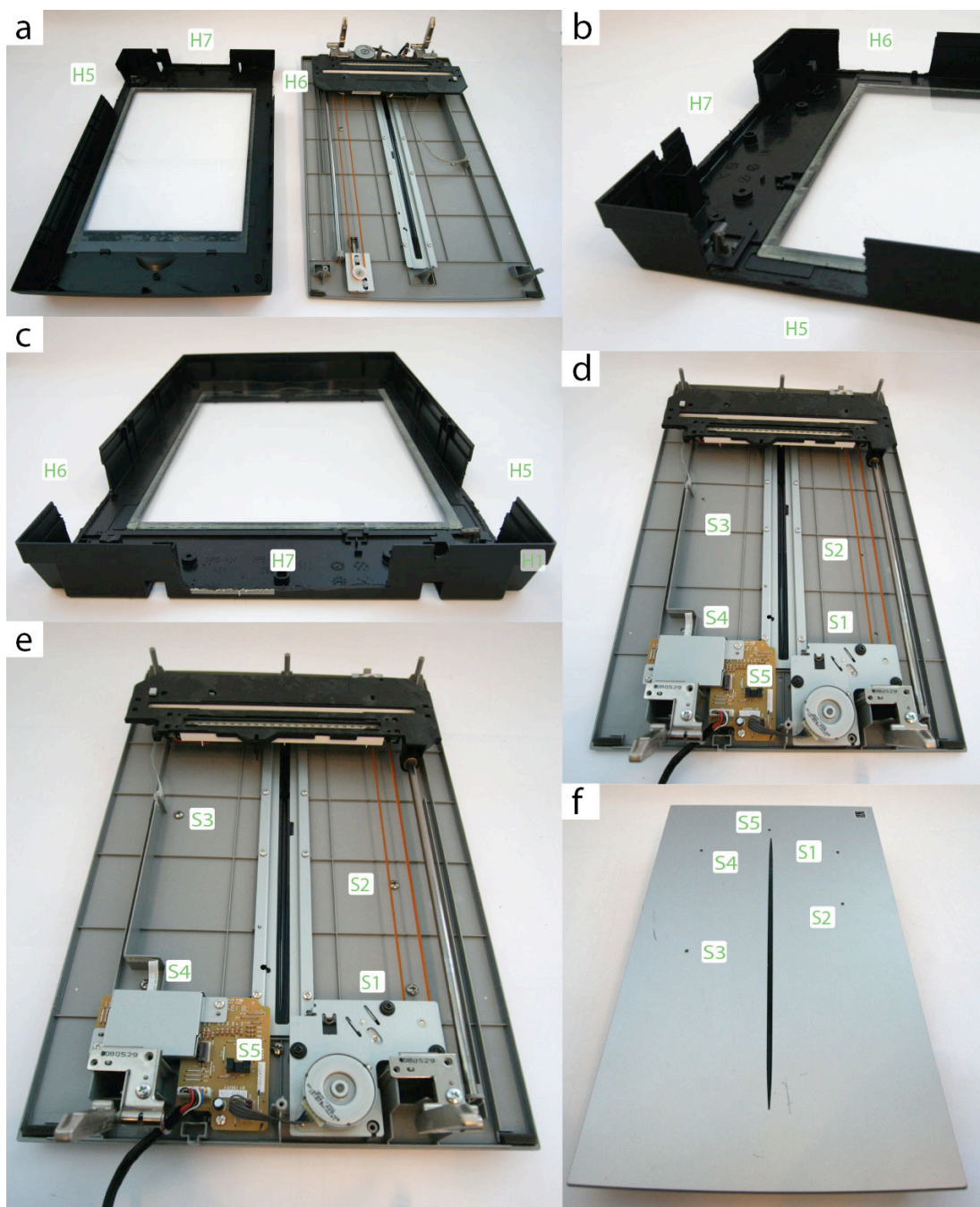


Figure 3-3: Photographs of mounted scanner fan. **(a):** The Transparency Unit (TPU) is shown with its top and bottom separated. **(b–c):** Three cuts, H5–H7, are made into the bottom of the TPU. **(d):** Four holes are drilled into the TPU lid, and **(e)** filled with machine screws. **(f):** The holes are shown from the top of the lid.

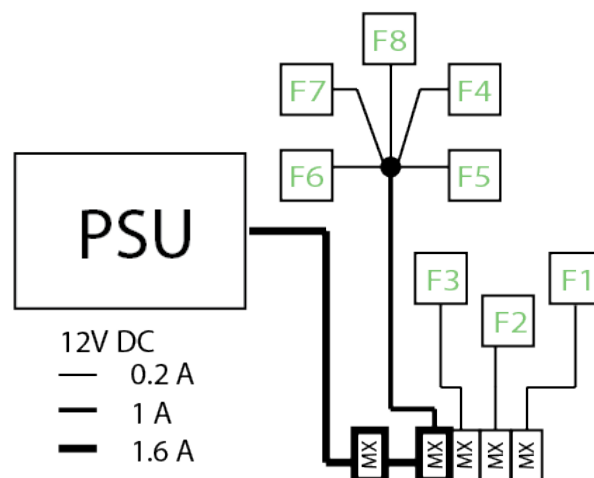


Figure 3-4: Fan wiring diagram. The wiring that supplies the fans is shown. “MX” indicates a standard Molex connector. The connection between fans F4–F8 can be made by clipping off the Molex connectors and splicing the wires together using solder, or by using a butt-clamp or a wire nut. Be careful to add appropriate insulation. Be careful to plug the Molex connectors in the correct order, otherwise excess current may be passed through each connector. Be careful to use an appropriate of gauge wire for the indicated current. The eight fans attached to each scanner draw about 1.6 amps at 12V DC. To prevent heat build-up along the wire, a voltage drop of less than 3% along the wire is widely considered safe. If the power source is 10 feet away from the scanner, this suggests that an AWG gauge no larger than 20 should be used. Incorrect wiring can generate excess heat and represents a fire hazard.

3.3.4 Accessing and preparing the fixed scanner lens

The Epson Perfection v700 contains a single fixed lens, whose position sets the focal plane of the device. Worms on agar plates are out of focus on stock scanners, because they are raised above the standard focal plane. To optimize image quality, we move the lens position very slightly.

- (1) The scanner lens is located inside the moving scanner bar, situated inside the bottom scanner chassis. Remove the lid of the scanner chassis (as described in Recommended Mounting Procedure Step 3), and identify the scanner bar. Refer to Figure 3-5.
- (2) There is a removable plastic flap in the top of the bar. Remove it with a screwdriver. Put it aside in a safe place. If you look into this slot, you should see, a few inches down, a black tube. This is the top of the lens. Unfortunately, scanners ship with this lens glued in place by a piece of hard resin. We need to remove this resin before we can adjusting the lens position. Removing the resin, in turn, requires we access the lens directly through a port on the bottom of the scanner bar. This, in turn, requires we disconnect the scanner bar from the scanner chassis, to make it possible to access the bottom port.
- (3) Identify the small metal clip that attaches the scanner bar to the drive belt. The drive belt is the long rubber belt running the length of the scanner chassis. Take a look at this clip and remember how it is attached. Pop it off by pushing down on the exposed lever.
- (4) Locate the nut at the front of the scanner that holds the support bar in place. The support bar is the long metal pole that running the length of the scanner chassis. Remove the nut. Notice that the bar is still held securely by a friction connection at the back of the scanner.

- (5) When handling the support bar, wear gloves to avoid exposing it to skin oils. Rotate the support bar in place to release the friction connection. Sometimes, pliers may be necessary. Only use pliers on the rectangular end of the bar, never on the cylindrical part. The scanner bar and support bar should now be loose
- (6) Remove the support bar from the scanner bar, and put it in a safe, clean place.
- (7) Flip the scanner bar so that rests upside down inside the chassis. Be careful not to catch any wires or to crease the ribbon cable.
- (8) You should see several screws on the bottom of the scanner bar, securing the bottom lid. Remove them, and remove the black bottom lid.
- (9) You should now be able to see the fixed lens, as shown in Figure 3-5. It is black, often with a white dot. There is a second lens, used for IR measurements, which you can safely ignore. You should see a little piece of gum melted onto the lens. Remove it with a screwdriver or a razor blade. This should take about a minute. You'll need to work around the edges, but usually chunks can be pried off in big pieces. Make sure to clean around the edges of the lens.
- (10) You have freed the lens! Replace scanner bar's bottoms cover, and secure all the screws. Make sure the ribbon cable running into the bar enters through the correct slot.
- (11) Slide the support bar back into the scanner bar, and fit everything back into its original place.
- (12) Rotate the support bar in situ such the friction joint holds it secure.

3.4 Instructions for focusing a scanner

- (1) Obtain a few reference plates with which to calibrate the focal plane. The authors' prefer to optimize focus using live worms on agar plates, but can imagine a more stable reference might be ideal. If live worms are used, let the plates dry enough so that they will absorb some liquid. Seed with 125–250 microliters bacteria and let the liquid absorb, providing the animals with a comfortable, dry, lawn.
- (2) Pick 20–40 animals onto each plate.
- (3) Very Important! Let the animals calm down for an hour or two at room temperature after plating, to ensure that they have stopped moving quickly. If calibration is attempted using startled worms, all images will appear blurry regardless of focus, due to animal motion. It is easy to waste an hour attempting, in vain, to focus fast-moving worms.
- (4) If the scanner chassis lid is currently removed, place it back on. Do not tighten any screws, because it will be necessary to enter the chassis in each step of the focusing process. Do make sure that the lid is seated flush. Sometimes, the lid can catch on one side, which will raise it slightly and affect the focus.
- (5) If the TPU is currently unattached, put it on and plug it in.
- (6) If the scanner isn't plugged in to a computer and a power outlet, do so.
- (7) If the chassis and TPU fans aren't running, plug them in and start them running. Scanning produces heat, and without operating fans, plates will begin to fog over during calibration.
- (8) Place the calibration reference plates into the scanner. If plates are to be placed on glass sheets during experiments (as is recommended), remember to include this glass sheet into the calibration process.
- (9) On the computer, run a preview scan that captures the entire scanner surface.
- (10) Select a small location of one plate on which to focus. Smaller areas will be captured more rapidly, speeding the calibration process. If worms are being used, try to select a small area that contains several worms. The calibration process is fastest considering a single plate, but this is somewhat risky as plates may vary slightly in height. Ideally, the focal plane should be adjusted to a consensus height that works well for most plates.
- (11) On the computer, capture an image.
- (12) Inspect the captured image at 100% zoom. The quality of the current scanner focus can be

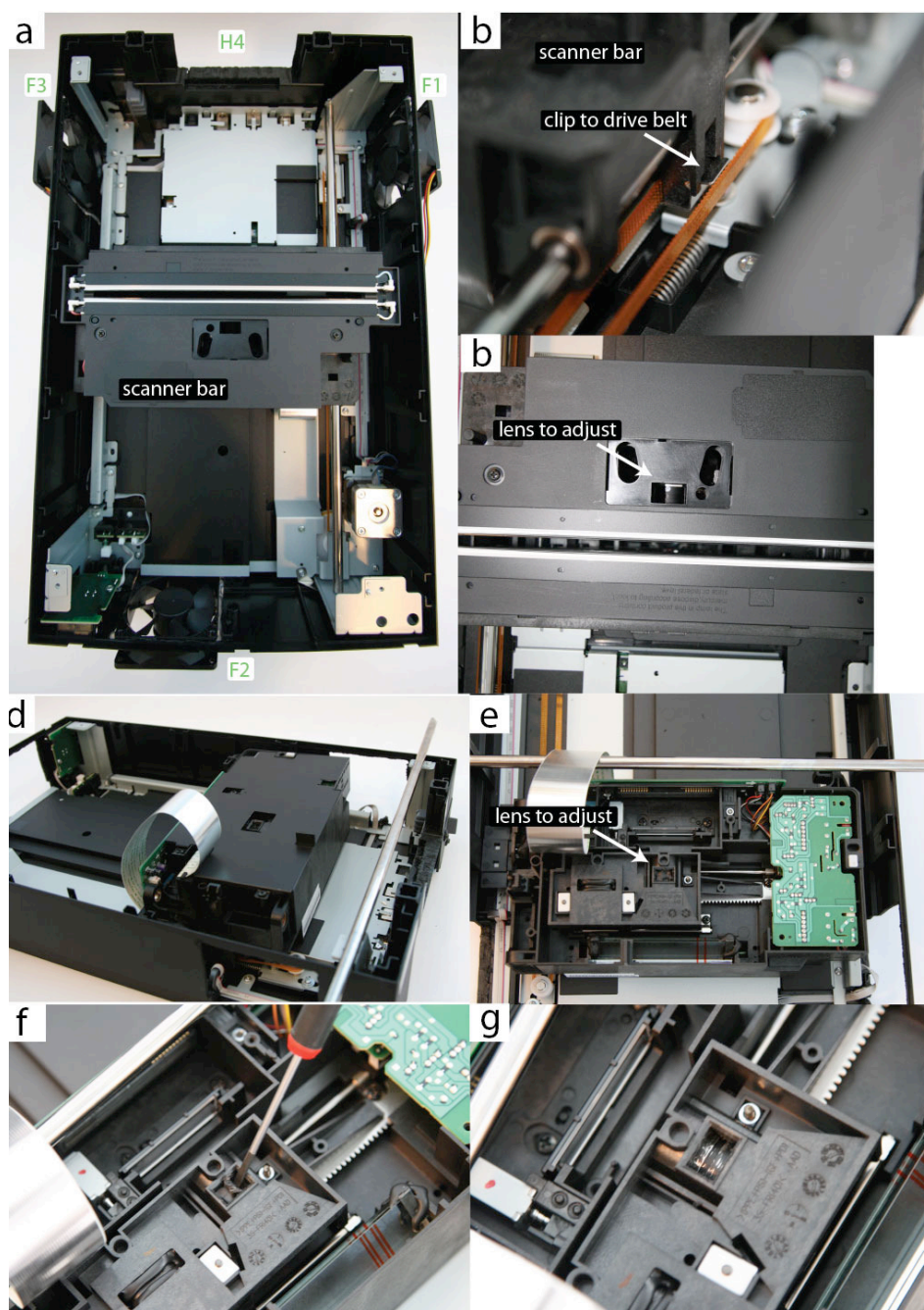


Figure 3-5: Inside the scanner chassis. **(a):** A top-down view of the scanner chassis with its lid removed. **(b):** A close-up of the clip that connects the scanner bar to the drive belt. **(c):** The slot in the scanner bar through which the lens can be adjusted. **(d):** The freed scanner bar can be overturned to reveal a bottom access panel. **(e):** With the panel removed, the bottom of the lens can be seen, glued in place. **(f):** The gum can be removed with a screw driver. **(g):** The lens freed from gum.

evaluated through comparison to a collection of gold-standard images, available in Figure 3-6.

(13) If optimal focus has not been achieved, remove the lid of the scanner chassis, along with the

entire TPU, as one piece. With proper organization of your workspace, it should be possible to do this without detaching any cables.

- (14) Look through the top access port of the scanner bar, and identify the fixed lens. With a small flat head screw driver, nudge the lens slightly in the desired direction. The lens should be adjusted in tiny, nearly imperceptible increments. If you think you've made a large adjustment, in all likelihood the scanner is now extremely out of focus. In this case, it will be a long job hunting your way back. One strategy is to scratch notches into the paint of fixed lens housing. These scratches can then be used as a reference to mark your position.
- (15) Reattach the chassis lid and TPU, making sure that the lid is seated flush.
- (16) Repeat steps 11–15 until a scanner is in focus. It is the authors' experience for half the scanners they have adjusted, optimal focus is achieved within two or three adjustments. In this case, focus is achieved in less than five minutes. For the other half of scanners, optimal focus is more elusive, requiring up to half an hour of iterations. Be patient, this process only needs to be done once in the lifespan of the scanner.

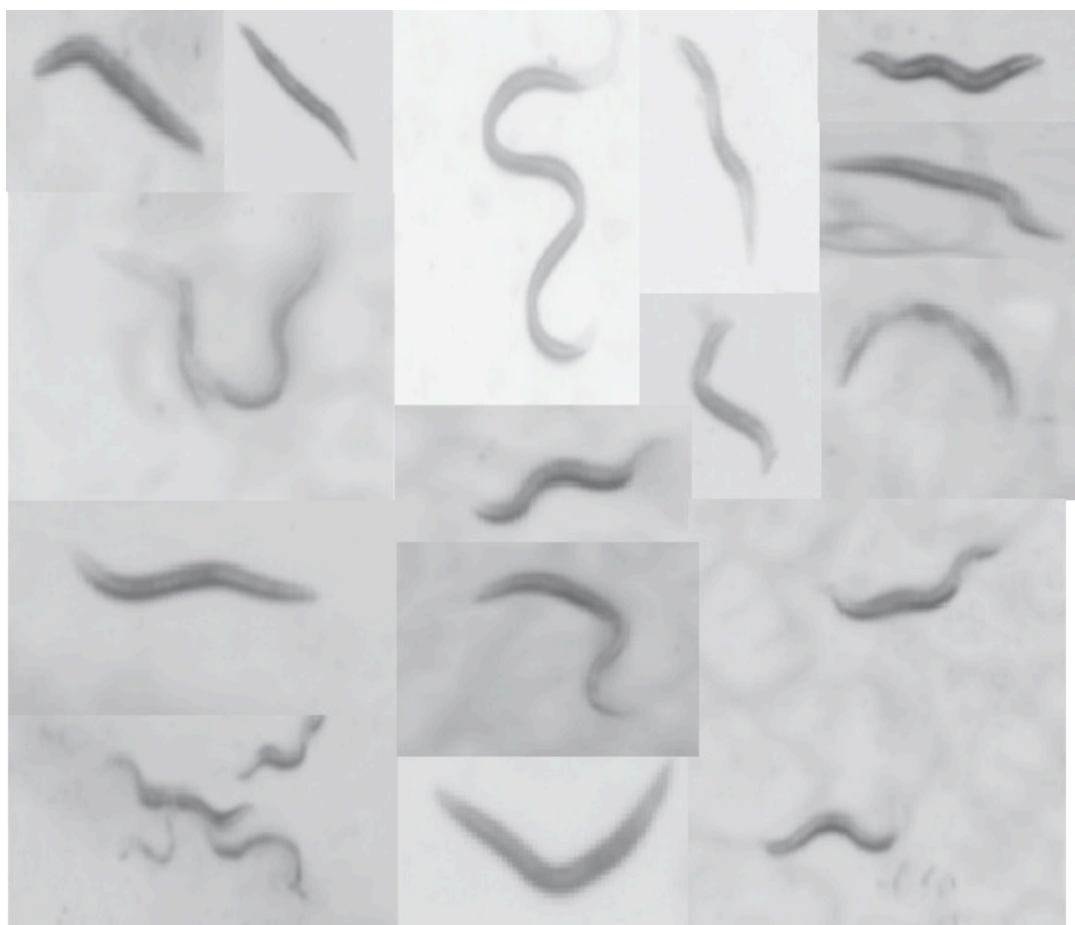


Figure 3-6: Reference images demonstrating sufficient focus. These images were collected across several scanners, and represent the end-point of scanner focal-plane optimization for the devices. In the authors' experience, attempts at acquiring images sharper than those presented are generally counterproductive.

Supplementary Note 4 — Software toolset

4.1	Software components	25
4.2	The Worm Browser GUI client	25

4.1 Software components

1. *Statistical software:* To assemble survival curves, we used JMP [3], a commercially available statistical package, and R [4], an excellent GPL-licensed statistical software environment. In R, we used the `surv`, `rms`, `flexsurv`, and `muhaz` packages.
2. *Worm Browser:* See section 4.2.
3. *Image acquisition server software:* Custom software controls experiments by communicating with the scanners via USB. Scanning schedules are stored in the MySQL database and take roughly the form “At time t , run a scan on device D between positions (x_1, y_1) and (x_2, y_2) ”. Typically, four plates are imaged in one scan, representing a column of plates on the scanner. The server monitors the schedule database and executes scans as they become due. Various telemetry is collected, including the start and stop times of each scan, various gross image properties such as average image intensity, and a description of delays or errors that occurred. Since a Lifespan Machine may run for three weeks or more, it is likely that errors will crop up in some system component, such as a scanner, the network connection, a power supply, etc. The image server software has several layers of redundancy and error handling to mitigate such problems, including the ability to send emails or text messages reporting new problems.
4. *Image-processing software:* The image processing pipeline and associated job scheduling software is written in C++. The Worm Browser is written in C++ using the library FLTK (GNU GPL V2) to provide cross-platform GUI functionality.
5. *Availability:* All software is currently (2013) accessible at www.lifespanmachine.org.

4.2 The Worm Browser GUI client

The Worm Browser is designed as a desktop client that complements the web interface to the lifespan machine. The Worm Browser performs five separate tasks:

- *Processing and uploading image capture schedules.* The Worm Browser accepts and processes a high-level description of an experimental schedule, written as an XML file. This schedule file contains a description of the image acquisitions that should be performed, in the style of “scan all plates on 10 scanners, executing a scan every 15 minutes. Do this for 28 days”. The worm browser compiles this file into a list of specific scan times, which it uploads to the central database.
- *Generating and processing mask files.* Individual plates must be identified in images of scans. The Worm Browser creates the mask files and saves them to local storage for annotation. The annotated mask file is then loaded back into the Worm Browser and the results submitted to the central database.
- *Inspecting and annotating worm image data.* The Worm Browser enables a user to browse through various representations of processed image data, including “storyboards” that contain images and videos of each identified animal as it goes through the death transition (Supplementary Video 6).

- *Output of statistical files.* The Worm Browser can produce a variety of statistical data germane to both the operation and the results of analysis.
- *Debugging image processing pipeline steps.* The Worm Browser acts as a platform for experimenting with new image processing techniques. This is often easier in a client than working with the image processing server directly.

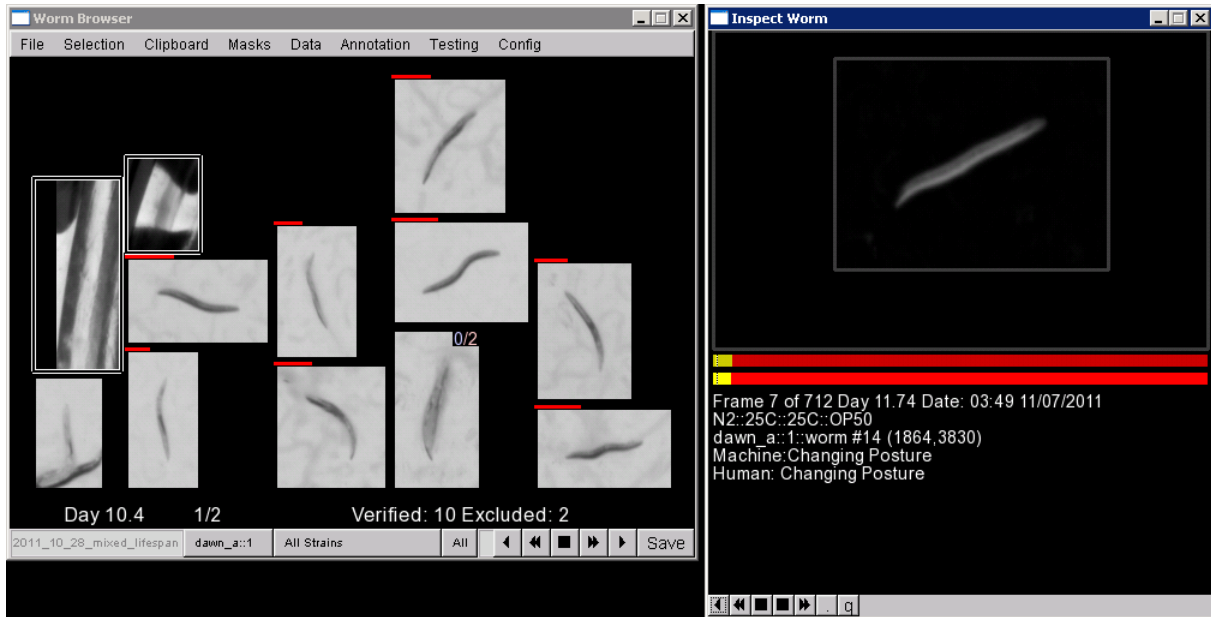


Figure 4-1: Worm Browser screen shot. See also Supplementary Video 6.

Supplementary Note 5 — Movement analysis

5.1	Identification of stationary worms	27
5.2	Posture analysis of stationary worms	29

5.1 Identification of stationary worms

Once foreground objects have been classified into worms and non-worms (Figs. 2-2 and 2-5), the task is to identify when individual worm-objects have permanently stopped moving. The Lifespan Machine distinguishes two kinds of motion: (i) spontaneous locomotion across the agar surface and (ii) subtle posture changes that do not involve whole body displacement (also referred to as *local* motion). Accordingly, the identification of death events occurs in two phases. First, we identify worms that have stopped locomotion. Such worms are termed stationary. Second, images of stationary worms are further analyzed to identify cessation of spontaneous posture changes. The retrospectively identified cessation of all motion is then recorded as a death event.

A trained Support Vector Machine (SVM), see Fig. 2-2 and Fig. 1d of the main text, identifies the position (x, y) of worm objects in each image $X^{(t)}$. Let

$$P^{(t)} = \{(x, y) \mid \text{object at position } (x, y) \text{ in } X^{(t)} \text{ is an SVM-identified worm}\}$$

denote the set of these positions. Early in a lifespan experiment, when most worms move around rapidly, the scanning rate of two images per two hour period does not allow determination of which worm in $P^{(t)}$ corresponds to which worm in $P^{(t+1)}$. To identify death events, however, we do not require such early-stage detail. It suffices that worms become trivially trackable once they have ceased locomotion. These worms are then monitored for subtle head and tail displacements, as described in Note 5 section 5.2. Identifying the onset of a worm's stationary phase is a retrospective analysis. It results in a straight "time-track" (Fig. 1d of the main text) with an associated series of images that is then subject to posture analysis. While this is straightforward in principle, the intermittent locomotion of worms, their spatial interactions, and the noisiness of image-capture (Fig. 2-1) and worm recognition create challenges.

We construct time-tracks of stationary worms in a stepwise fashion and clean them up prior to posture analysis. The time-track of a stationary worm w_i is defined as a sequence of locations:

$$w_i = [p_1 p_2 \cdots p_n]_i \text{ with } p_1 \in P^{(t_1)}, p_2 \in P^{(t_2)}, \dots, p_n \in P^{(t_n)}, t_1 < t_2 < \cdots < t_n \quad (1)$$

The time-ordered series of stationary positions $p_k \in P^{(t_k)}$ are meant to correspond to the same spatial location across images $X^{(t)}$. The time points need not necessarily be consecutive scans, i.e. the track may contain gaps. Such gaps can arise for several reasons, such as the temporary misidentification of the object at that position (a second worm might transiently cluster or cross, creating a structure not recognized as a worm) or the behavior of the tracking algorithm (below). These time-tracks are built up sequentially proceeding backwards from the last scan in the experiment toward earlier scans. Stationary tracks of worms that died early may not remain identifiable throughout the experiment, since a dead animal may dissolve over time and become unrecognizable by the SVM. Thus, despite proceeding backwards from a state in which all worms are dead, we might nonetheless have to initiate new stationary tracks as we encounter their ending.

Outlining the procedure requires a bit more notation. Since a worm's time-track is built over time, we decorate w_i , defined above, with a scan time t , i.e. $w_i^{(t)} = [p_1 p_2 \cdots p_n]_i$, to indicate the most up-to-date time-track as of time $t \leq t_1$. (Remember we proceed backward in time; at the next addition of a time point, the current p_1 becomes p_2 and the current t_1 becomes t_2 .) To denote the *set*, rather than the *sequence*, of positions belonging to worm i , we write $\text{Set}(w_i^{(t)}) \equiv \{p_1, p_2, \dots, p_n\}_i$.

This allows us to express that no location can belong to more than one time-track: for all $i \neq j$, $\text{Set}(w_i^{(t)}) \cap \text{Set}(w_j^{(t)}) = \emptyset$. We refer to the most recently added component (youngest animal age) of the time-track of worm i as $w_{i1}^{(t)}$, that is, $w_{i1}^{(t)} = p_1$ of $w_i^{(t)}$. Throughout the process we maintain a set of time-tracks $W^{(t)} = \{w_i^{(t)}\}$, each track tentatively belonging to a single worm. Corresponding to $W^{(t)}$, there is a set of anchor points $W_1^{(t)} = \{w_{i1}^{(t)}\}$ containing the most recently identified points of each track. Finally, we maintain a set of positions that have not been assigned to a time-track retrospectively up to time t , $F^{(t)} = \{p \mid p \in P^{(\tau)}, \tau \geq t \text{ and } \forall i \ p \notin \text{Set}(w_i^{(t)})\}$.

Given sets $W^{(t)}$ and $F^{(t)}$, we need to identify which positions $P^{(t-1)}$, derived from scan $t-1$, extend (backwards in time) existing time-tracks in $W^{(t)}$, or initiate a new time-track. This is accomplished by finding an assignment $f : W_1^{(t)} \cup F^{(t)} \mapsto P^{(t-1)}$ that minimizes the total Euclidean distance $\|\cdot\|$ between positions p_k in $W_1^{(t)} \cup F^{(t)}$ and their images under f :

$$\sum_k \|p_k - f(p_k)\| \stackrel{!}{=} \text{minimum} \quad (2)$$

where as many elements as possible must be assigned. If either $W_1^{(t)} \cup F^{(t)}$ has fewer elements than $P^{(t-1)}$ or vice versa, some positions must be left unmatched (i.e. they are matched to nil). This combinatorial optimization problem is of a well-known form that can be solved [5].

The new set $F^{(t-1)}$ (recall that we proceed from later towards earlier times) is formed from all elements of $F^{(t)}$ that were not matched to a $p \in P^{(t-1)}$ plus all $p \in P^{(t-1)}$ that have no match in $W_1^{(t)} \cup F^{(t)}$. Each element in $F^{(t)}$ paired with an element in $P^{(t-1)}$ initiates a new time-track (and thus identifies a new worm) that is added to $W^{(t)}$ in producing $W^{(t-1)}$. A $p \in P^{(t-1)}$ that pairs up with $w_{i1}^{(t)}$ extends the time-track of worm $w_i^{(t)}$, i.e. $w_i^{(t-1)} = pw_i^{(t)}$. In effect, p becomes the new $w_{i1}^{(t-1)}$. Gaps occur when a freshly paired $w_{i1}^{(t)}$ is several scans old. The whole process begins with $t = T$ (the last scan), setting $F^{(T)} = P^{(T)}$, $W^{(T)} = \emptyset$ and, consequently, $W_1^{(T)} = \emptyset$.

The mechanism advancing the scanning bar causes jitters of a few hundred microns that make it difficult to exactly map position (x, y) in one image of a plate to the same position in a subsequent image (Fig. 2-1). Thus, for two positions to correspond to each other, we require them to be within distance r . In optimizing the assignment f , we therefore exclude (penalize) pairings between a position in $W_1^{(t)} \cup F^{(t)}$ and a position in $P^{(t-1)}$ whose Euclidean distance is larger than r . This speeds up the optimization process and tunes it to detect stationarity, since pairings between more distant positions are unlikely to refer to the same animal in our time-lapse context.

Various sources of noise and measurement error may produce non-optimal behavior of the locomotion analysis procedure. For example, $w_i^{(t)}$ may drift perceptively over time, as plates dry slightly or the scanner bar drifts. The r -criterion may allow two time-tracks to drift close to each other and “cross over”, or failure of the SVM model to detect worms may produce time-tracks with excessive intermittency or very long gaps. Spurious events may be introduced by fast moving worms that are coincidentally caught close to the anchor point of a worm time-track. To handle all of these cases, paths are discarded or merged (“cleaned up”) according to a panel of criteria, prior to being submitted to local movement analysis.

- A w_i is discarded if it is too short (total duration of less than 6 hours), too sparse (i.e. it contains points at only half of all its observation times), or it drifts quickly (the average distance between consecutive scans is greater than 5 pixels).
- If two or more w_i are located very close to each other (less than 50 pixels) near the beginning

of one w_i and near the end of the other, it is likely that a single stationary animal was subjected to a one-time displacement, causing it to be incorrectly split into two w_i . If the corresponding ends of the two w_i s do not overlap for a very long time (more than two hours) and are not separated by a large gap in time (more than twelve hours) the two w_i s are merged.

The end result of this analysis is a set of time-tracks, as shown in Fig. 1d of the main text. Each straight track corresponds to a stationary worm with an associated stack of cropped images.

5.2 Posture analysis of stationary worms

The input to local movement analysis is a series of cropped images $X_i^{(t)}$, $t = 1, \dots, T$ of a stationary worm i , with $t = 1$ indicating the image from the first scan and $t = T$ the image from the last scan. For the sake of less clutter, we shall omit the index i with the understanding that the following analysis pertains to a single time-track.

All images have undergone background subtraction as described in Note 2 section 2.1. Furthermore, a scanner does not produce perfectly aligned images in consecutive measurements of the same plate (Fig. 2-1). Each image $X^{(t)}$ is therefore aligned computationally with the consecutive image $X^{(t+1)}$ to remove machine jitter. An image is a set of pixel intensities on an (x, y) -grid and represents a discrete sampling of the continuously varying intensities being measured. Although pixels are located at integer offsets, the scanner alignment errors resulting from the advancing scanner bar can vary continuously. To account for such sub-pixel errors in alignment, we subsample images to allow registration offsets of distances less than a single pixel.

A typical stationary worm series with posture changes is shown in Fig. 1e of the main text. The objective of local movement analysis is to detect these changes by comparing consecutive images of the same stationary worm. Since animals may take a variety of postures, and various mutants may present novel postures, we do not attempt to determine the exact form of these changes. Instead, we simply determine the overall change Δ_t in pixel intensity inside the alignment frame between $X^{(t)}$ and $X^{(t+1)}$. This strategy also has the advantage of not relying on accurate segmentation, allowing subtle head movements to be detected even in the common situation in which heads are not fully retained during segmentation.

Consecutive images of the same plate region may vary in overall intensity simply because of variations in the output of the scanner light source or stray light. To compensate for differences in average brightness between subsequent images, we use as the local movement score

$$\Delta_t = \sum_{i=1}^N |(x_i^{(t)} - \overline{X^{(t)}}) - (x_i^{(t+1)} - \overline{X^{(t+1)}})| = \sum_{i=1}^N |x_i^{(t)} - x_i^{(t+1)}| - \left| \sum_{i=1}^N x_i^{(t)} - \sum_{i=1}^N x_i^{(t+1)} \right| \quad (3)$$

where $\overline{X^{(t)}}$ denotes the mean pixel value in image $X^{(t)}$ and the summation runs over all N pixels in the alignment of images $X^{(t)}$ and $X^{(t+1)}$. The above expression yields a time series of local movement scores Δ_t , $t = 1, \dots, T - 1$ for a stationary worm (see Fig. 1e in the main text and Figs. 7-1a, 7-1b, 7-3c).

Additional forms of scanner noise may cause Δ_t to deviate from zero even for dead worms (Fig. 1e in the main text and Fig. 7-2b). This necessitates a one-time calibration step to define a threshold value Δ_{\max} below which we consider an animal as non-moving. However, dead worms might appear to sporadically move as live animals push them around. We therefore define an animal as *permanently* non-moving, i.e. dead, when it is observed below Δ_{\max} for longer than a threshold value τ (which exceeds the maximum resting time of live animals). To obtain these two threshold values, Δ_{\max} and τ , we assemble a “gold standard” time series for several hundred worms whose

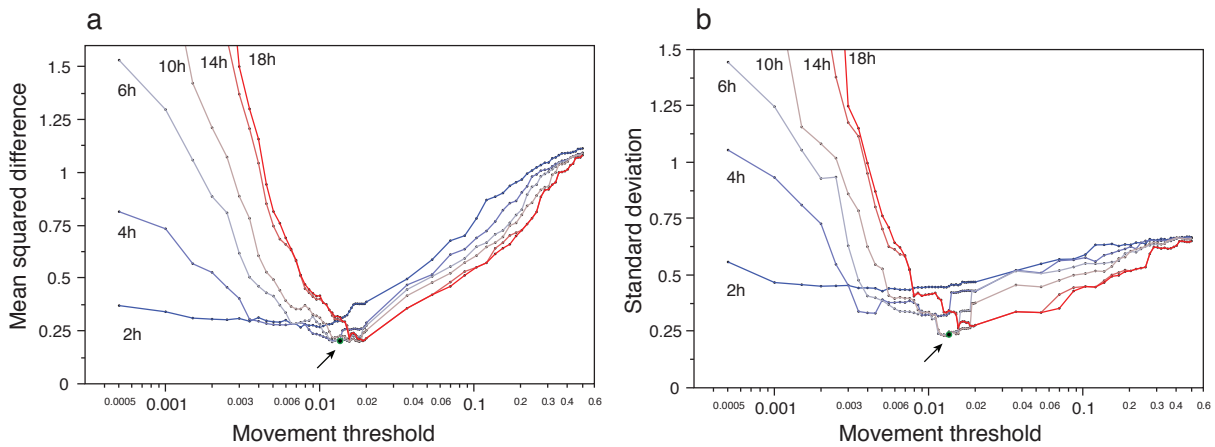


Figure 5-1: Parameter robustness of posture analysis. To determine an optimal combination of the parameters Δ_{\max} (movement threshold) and τ (hold time), image strips of several hundred animals were inspected using the Worm Browser and their death times were annotated. Many combinations of threshold and hold time values were systematically tested. Each combination was used to estimate a death time for each animal using our posture analysis algorithm, resulting in an estimation error when compared to the by-hand annotation. The panels show the mean squared error **(a)** and the standard deviation of the squared errors **(b)**. The algorithm produces accurate and robust results for threshold values between 0.01 and 0.02, and for hold time values between 6 and 14 hours. The chosen parameter combination is shown as a green dot (arrow).

death times are annotated by visual inspection of the image stack in the Worm Browser (Note 4 section 4.2 and Supplementary Video 6). The thresholds are chosen such as to minimize the deviation between computed and certified death times of the gold standard set, Fig. 5-1. Fig. 5-1 also indicates that the death calls are fairly robust with regard to the so-chosen parameters. Using these parameters, the automated analysis of posture changes returns the first t at which Δ_t last falls below threshold for the required duration of time. The discrete time interval associated with the death event is $[t, t + 1]$, and corresponds to the wall-clock time elapsed between onset of adulthood and scans t and $t + 1$.

We examined several image transforms, such as equalization or normalization, prior to pairwise comparison of images that might amplify and clarify local motion. Yet, none yielded a distinctive advantage over the simple brightness adjustment, equation (3).

Supplementary Note 6 – Statistics of worm movement

The movement analysis described in sections 5.1 and 5.2 can be used to partition the population of worms into movement classes, similar to those defined in [19]. Stationarity (or posture) span refers to the time between cessation of translatory locomotion and death, when the only motion consists in subtle changes of head and tail position without affecting overall location. For wildtype and *age-1(hx546)* animals shown in Fig. 3 of the main text, we calculated the stationarity span and find that wildtype animals spent 1.27 ± 0.17 days, whereas *age-1(hx546)* mutants spent 3.0 ± 0.15 days in this state, see Fig. 6-2.

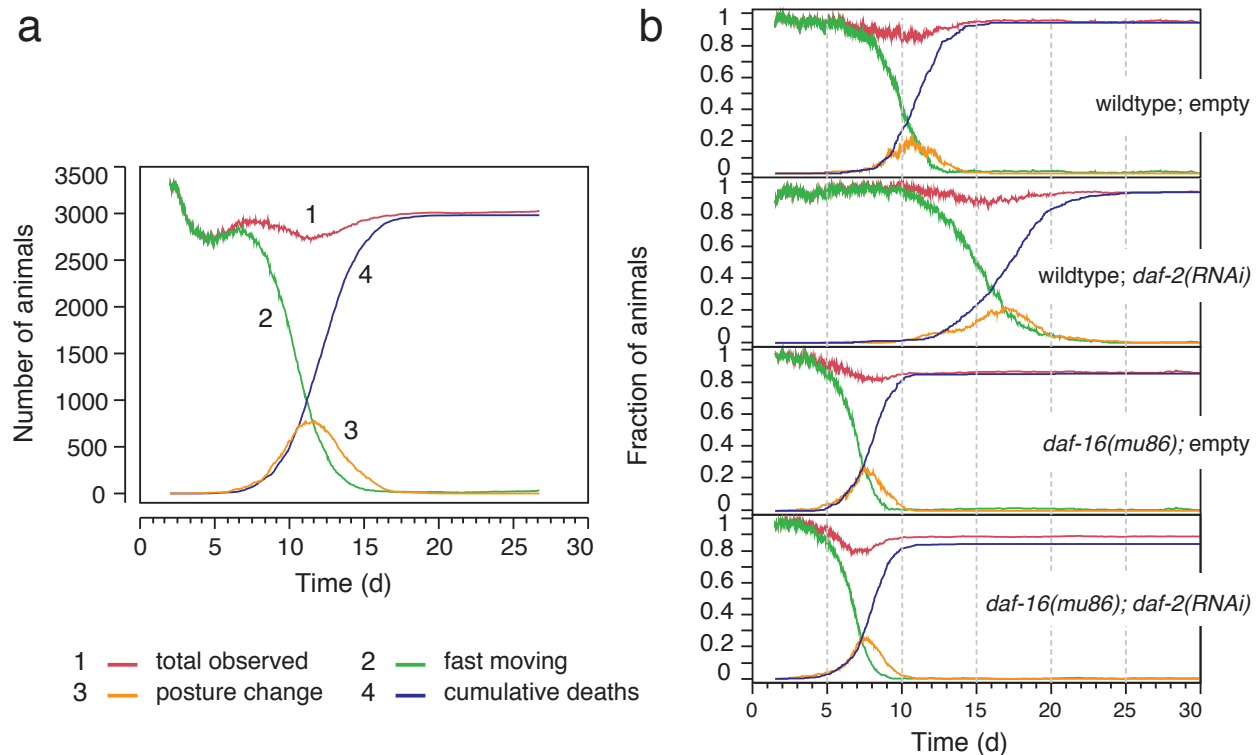


Figure 6-1: Population statistics of movement classes. **(a):** The red curve (curve #1) reports the total number of wildtype worms observed by the machine at any time in this experiment. (This number changes for reasons detailed in Note 8 section 8.2 and Fig. 8-3). The green curve (#2) is the number of individuals that crawl across the agar, i.e. are not stationary in the sense of Note 5 section 5.1. The orange curve (#3) shows the number of worms that are stationary, but still change posture in the sense of Note 5 section 5.2. The blue curve (#4) reports the number of animals that have died. The overall plot exhibits a population dynamics of movement categories reminiscent of an irreversible chemical reaction chain in which $A \rightarrow B \rightarrow C$. **(b):** As in panel (a), but for wildtype and *daf-16* backgrounds with *daf-2* RNAi and empty vectors.

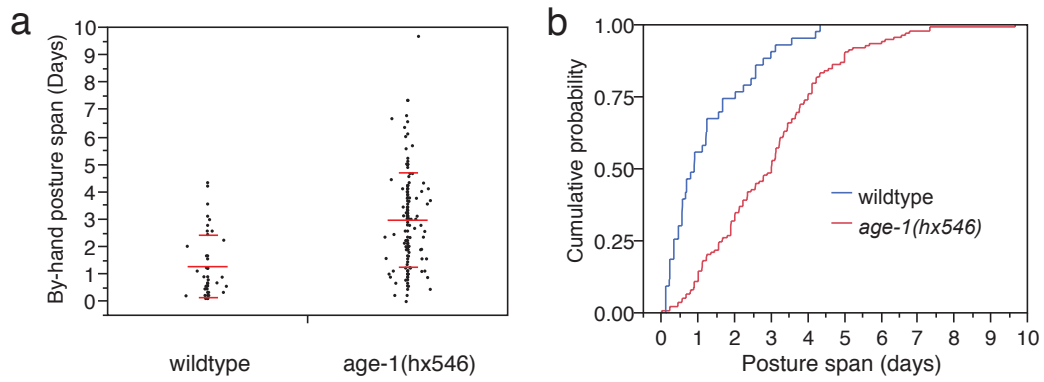


Figure 6-2: Stationarity (posture) span. **(a)** The time spans between cessation of translatory motion and death (“stationarity spans”) are shown with overlaid averages (wide bar) and standard deviations (narrow bar). **(b)** Comparison of the cumulative stationarity span distributions. 1.0 minus this curve would be a “survival curve” whose terminal event is death and whose initial time point $t = 0$ corresponds to cessation of translatory motion.

Supplementary Note 7 – A “death phenotype”

After an animal ceases to crawl, the image analysis of the Lifespan Machine produces a time series of registered images of that animal. This image stack, typically used to quantify local movement and identify death times (Note 5 section 5.2), can also be used to longitudinally monitor morphological

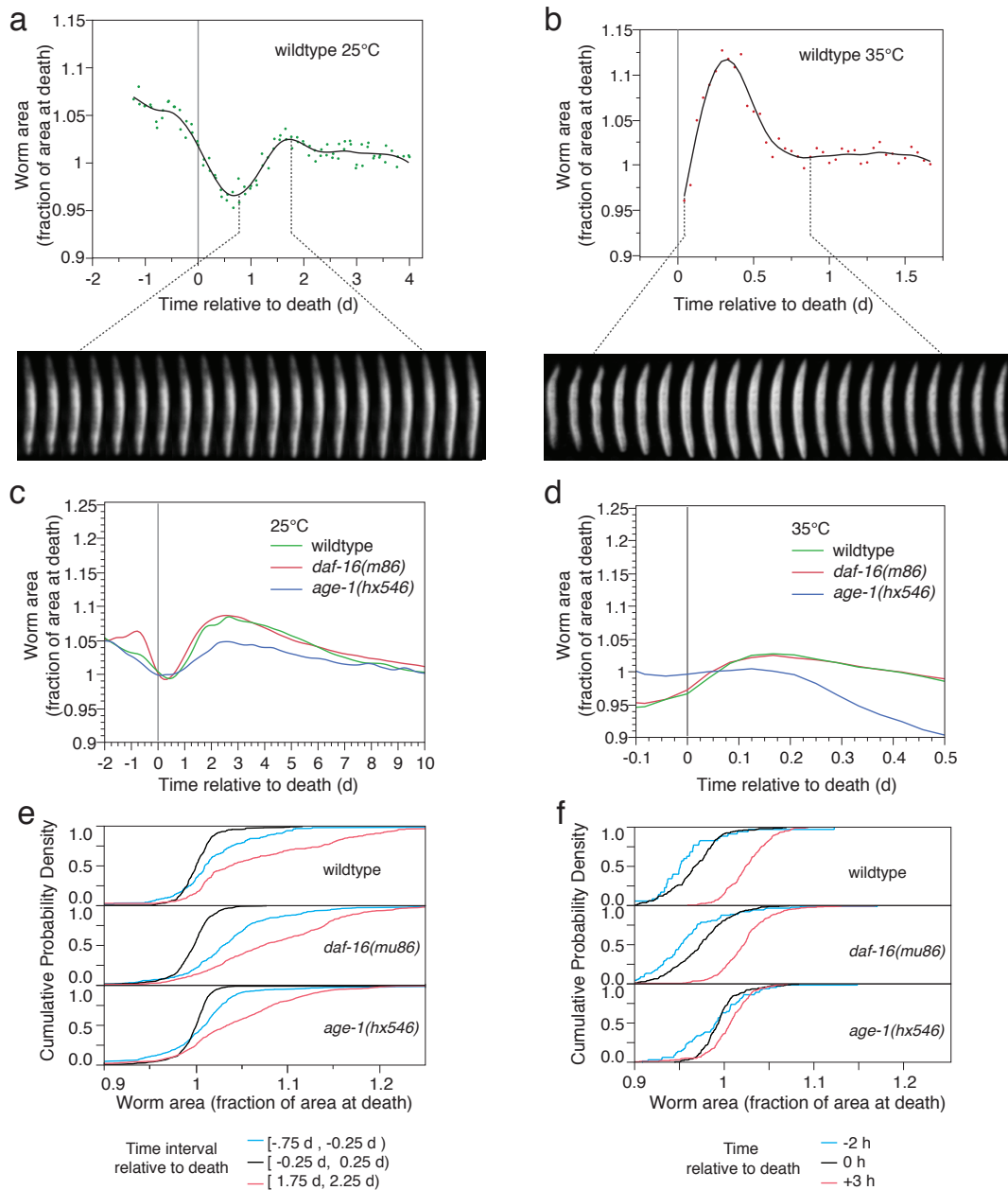


Figure 7-1: The volume of a worm dips at or near death. **(a)** We followed the total area (in pixels) of a single wildtype animal dying on OP50 at 25 °C. This area is plotted against time before and after the animal’s death ($t = 0$) as determined by visual inspection. Dots represent scores; solid lines are cubic spline fits [3, 18]. The size of the animal decreases starting about a day before its apparent death, after which the trend sharply reverses and size increases. The volume of a worm dips at or near death. (Figure on previous page.) **(b)** A single wildtype animal dying at 35 °C exhibits a consistent increase in size after its death. Caption continued on next page.

Figure 7-1 (Cont'd): Data prior to death is scant, because the stationary movement phase at this temperature is very short. **(c)** To evaluate whether death-related contraction and expansion of worm area is typical, we determined individual time-series of areas for 44 wildtype, 90 *daf-16(mu86)*, and 143 *age-1(hx546)* animals whose death time on OP50 at 25 °C was determined by visual inspection of the image stack using the Worm Browser. A cubic spline was fit to minimize the mean squared error across all individuals at each measurement. This reveals a conspicuous contraction before death and expansion thereafter. **(d)** We proceeded as in (c) for 189 wildtype, 304 *daf-16(mu86)*, and 194 *age-1(hx546)* animals on OP50 at 35 °C. Animals dying from acute heat stress were not observed to show a decrease in size immediately before death. This either represents a true difference in late-life morphological changes compared to animals dying at 25 °C, or the shrinking close to death occurs while animals are still crawling and can, therefore, not be measured with this technique. **(e)** The average behavior of worms shown in (c) and (d) may mask individual variation. To estimate the effect of such variation, we grouped all measurements made within three half-day intervals—the first between 0.75 and 0.25 days before death, the second within ± 0.25 days of death, and the third between 1.75 and 2.25 days after death. The cumulative distribution of each group is shown for each genotype. In all cases, the middle interval (± 0.25 days surrounding death) showed the smallest median and mean size, indicating that animals reach their smallest area just as they cease to move. **(f)** At 35 °C, animal area was considered at three single time points, two hours before apparent death, at apparent death, and three hours after apparent death. Animals appear to begin increasing in size before their last movement, and the size continues to increase afterwards.

features of worms nearing death. This led us to discover that most animals exhibit a stereotyped morphological change at or near the time of their final posture change. They first shrink by 10% or more and subsequently expand by more than that amount. The phenotype is established in Fig. 7-1, its coincidence with the cessation of all spontaneous motion (the death criterion of our method) is shown in Fig. 7-2, and Fig. 7-3 indicates its generality.

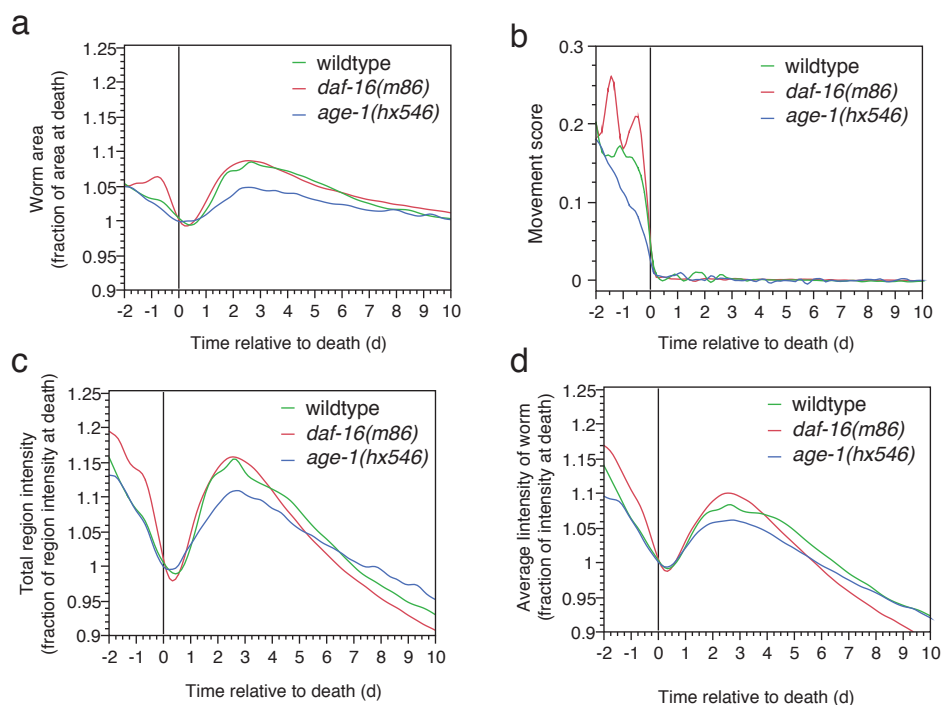


Figure 7-2: Coincidence of volume dip and cessation of motion. Caption on next page.

Figure 7-2 (Cont'd): Coincidence of volume dip and cessation of motion. **(a)** This is the same panel as in Fig. 7-1c. **(b)** To corroborate that these volumetric changes occur contemporaneously with the cessation of movement, we plotted spline fits (as in Fig. 7-1) of the corresponding movement score time-series. Death time identification by movement score is seen to agree with visual determination of death. On average, animals indeed cease to contract and begin to expand approximately at the time of their last movement. **(c)** On visual inspection, animals appeared to darken slightly with size. Pixel intensity of animals and animal area might be related, since area is calculated from a segmented image, whose segmentation depends on pixel intensity. To demonstrate that worms exhibited a net decrease in pixel intensity independently of segmentation, we measured the total pixel intensity within a box of constant size bounding the animal across its image time series. This metric shows the same trend as worm area, providing further evidence that morphological changes surrounding the death time are independent of our segmentation algorithm. **(d)** We determined the average intensity of pixels inside worms over time. This metric, too, shows the same trend as worm area, suggesting that the animals both shrink and diffract less nearing death, while getting larger and darker post mortem.

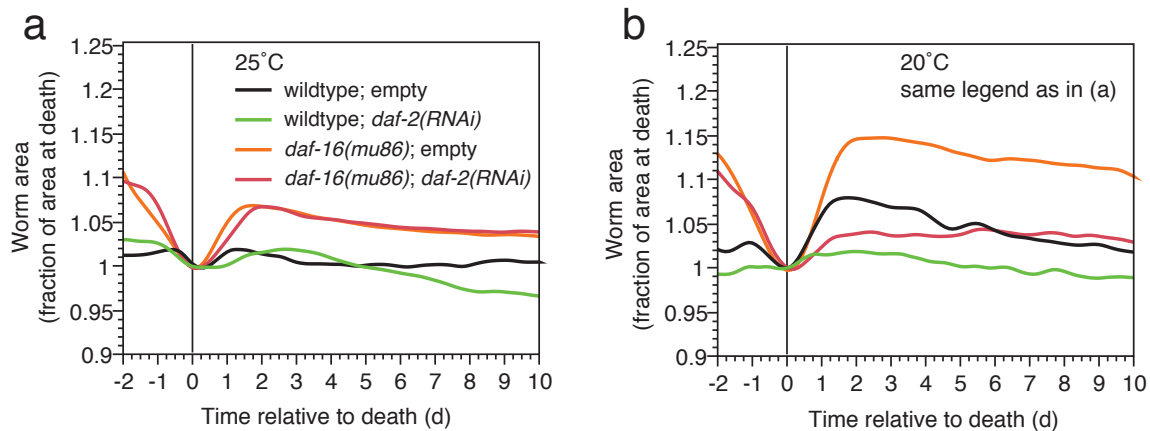


Figure 7-3: Generality of the volume dip. To assess the generality of the late-life characteristics described in Fig. 7-1, we performed the same analysis on animals grown on a different bacterial food source, HT115. **(a)** Area vs time relative to death on HT115 at 25 °C. **(b)** Area vs time relative to death on HT115 at 20 °C.

Supplementary Note 8 – Censoring

8.1	Right censoring and interval censoring	36
8.2	Tracking population size and right censoring strategies	37
8.3	Worm clusters and interval censoring	40
8.4	Censoring strategies for worm clusters	41
8.5	Determining total population size	42

8.1 Right censoring and interval censoring

Individuals are “lost” from a lifespan experiment when supervening events preclude the observation of the age-induced death of these individuals. For example, in both manual and automated lifespan experiments, worms are occasionally found dehydrated on the plate wall. These individuals are lost from further observation, but cannot simply be ignored during statistical analyses, as doing so would involve consideration of a biased sample. Kaplan and Meier proposed a nonparametric method for estimating survival curves that takes into account the contribution of individuals to the population at risk up until the moment they are lost [7]. We compile our survival curves according to Kaplan-Meier (KM).

Many statistical packages, like JMP, require time-to-event data to assemble a survival curve.

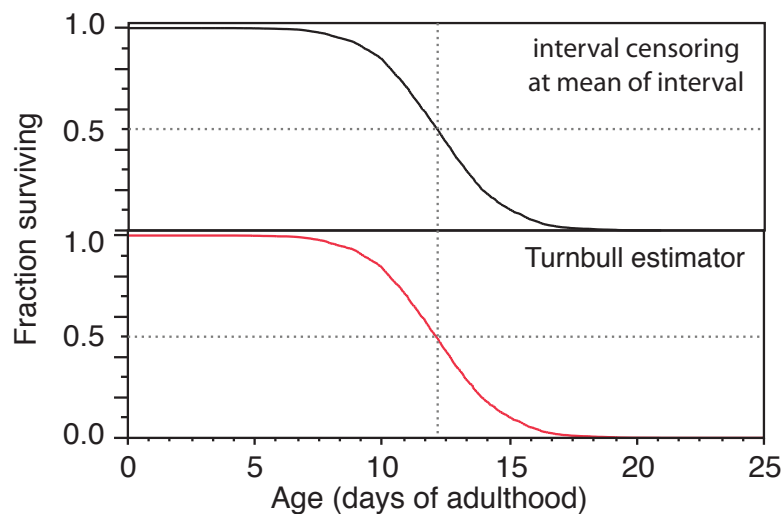


Figure 8-1: Interval midpoint vs Turnbull estimator. See text for details. The two curves are effectively indistinguishable if drawn on top of each other.

Our scans, however, happen at discrete times. Death or loss events are therefore only known to occur within a time interval defined by successive scans. Based on our scanning schedule, these intervals are typically either short (15 minutes) or long (1 hour and 45 minutes), but they can be longer if the system fails to detect a worm in scans immediately preceding its death. Quite generally, when events are only known to have occurred within an interval, a number of approaches exist to handle what is generally referred to as “interval censoring”. One interval censoring technique is the Turnbull estimator [8], provided by statistical packages such as JMP [3] or STATA [9]. When scan intervals are small compared to even the shortest lifespan in the population, a reasonable and computationally much faster alternative consists in simply choosing the interval midpoint as the time of death for all events known to have occurred in that interval, and then apply standard time-to-event KM.

We compared the Turnbull estimator with the simple midpoint estimation (Fig. 8-1). Using Turnbull estimation, the median of survival fell between 12 days 3.53 hours, and 12 days 3.60 hours. Using the interval midpoint, the median of survival occurred at 12 days 3 hours. This 36 minute difference is overwhelmed by several sources of experimental error that can produce much larger differences, for example temperature. In view of this negligible difference, the substantially faster speed, and the more useful analytics provided by many statistical packages when using time-to-event KM, we chose the midpoint estimate as our standard practice. All our data, however, are optionally supplied in an interval format suitable for Turnbull estimation.

8.2 Tracking population size and right censoring strategies

As in the manual lifespan procedure, worms whose lifespans are acquired automatically can be lost before their death is observed. Such animals must be right censored when their observation ends. A classic loss occurs when a worm crawls up the vertical sidewalls of a plate. In the Lifespan Machine, this causes a decrease in the total count of worms observed on that plate. Other events that require censoring are of biological nature, such as the “rupture” of a worm that fails to release fertilized eggs. These events are currently not detected automatically, and are counted as deaths, unless explicitly censored by visual inspection with the Worm Browser during the quality control phase that follows data acquisition and image processing.

One source of censoring events is specific to the automated method. Individuals can aggregate into clusters or clumps of two or more animals, where they are usually either misidentified during image processing as a single worm or ignored entirely (Fig. 2-5). Yet, for living animals, the resultant loss is often only temporary, as these individuals become again observable once they leave a clump. Dead animals inside clumps, as well as animals that leave the field of observation will remain missing permanently. We thus refer to an animal that is temporarily unaccounted for as *missing*, to distinguish it from being *lost* (i.e. “permanently missing”). For the purpose of adjudicating (right) censoring events, a worm inside the Lifespan Machine can therefore be in three states—missing, lost, or dead. A significant number of animals may be missing at times. In the manual procedure, observers can search for missing animals and physically separate clumps, but the Lifespan Machine has no such options. This difference requires additional processing to handle censoring. First, we must track the total number of animals on a plate from which we can infer missing animals. Second, we must determine how many among the missing animals were lost (i.e. never returned from the missing state) and at what times these losses occurred.

Because young worms move large distances between consecutive observations, we cannot directly determine when a particular animal goes missing nor, in case of its return, the time when it becomes observable again. Instead, we apply a tunable heuristic, explained below, to estimate the timing of both events. To estimate the statistical effects of this heuristic on our survival curves, we compare two extreme cases that delimit whatever unknowable histories have occurred in an experiment. The curves derived from these extreme censoring protocols are compared to each other in Fig. 8-5.

Let t index scans. Let $d(t)$ denote the number of deaths identified at time t and $D(t)$ the number of deaths that have occurred up until (and including) t , $D(t) = \sum_{i=0}^t d(i)$. Further, let $a(t)$ be the number of worms alive at t and N the total number of worms present initially. At any given time a worm is either alive or it has died sometime in the past (including now) or it is missing:

$$a(t) + D(t) + m(t) = N \quad (4)$$

with $m(t)$ denoting the number of worms that are unaccounted for (missing) at time t . Automated analysis of the image stack acquired during the lifespan experiment yields $a(t)$ and $D(t)$, providing us with $m(t) = N - a(t) - D(t)$. Worms mostly go missing in three ways: (i) A scan does not image

a worm properly. (For example, it may have crawled near or up the vertical plate wall.) (ii) A worm is misidentified as a non-worm by the classifier. (iii) A worm has disappeared into a clump of worms where it goes undetected. In all of these cases, a worm may return from the missing and be accounted for at a later time. Therefore, a plot of $m(t)$ versus t typically results in a non-monotonic step function, a schematic of which is depicted in Fig. 8-2 (green curve) and an actual example is shown in Fig. 8-3b.

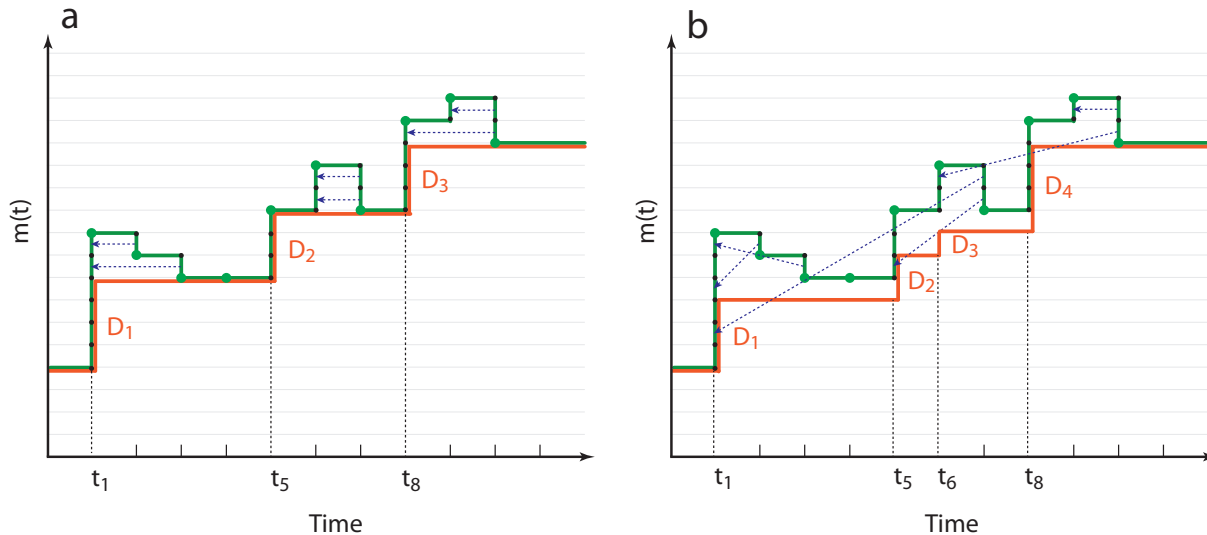


Figure 8-2: Going missing and getting lost (schematic). In both panels, the green curve represents a schematic $m(t)$ of equation (4), the number of animals detected missing over time. Each up-step (in the direction of increasing time) indicates the number of animals that have joined the ranks of the unaccounted for. Each down-step indicates the number of those that have become observable again. The orange curve is the censoring curve. Each up-step indicates the number of animals lost from further observation and that must therefore be right censored at that time interval. **(a):** In this scenario, the orange curve is constructed right-to-left, matching the green curve at first. Each green down-step of size 1 is matched against the closest up-step of size 1 and the orange curve is extended to the green level at that last match. **(b):** In this scenario, each green down-step is matched at random with a previous green up-step. When all down-steps have been paired, the orange curve is built left-to-right, tracking the number of unpaired green up-steps. Although the green curve $m(t)$ is the same in cases (a) and (b), the two strategies result in quite different censoring curves.

When $m(t)$ decreases, some worms that went missing before t are again accounted for. Yet, we have no information to deduce when these recaptured animals went missing. In other words, we cannot determine unambiguously how many of the worms that went missing at a given time were actually lost. Consequently, we cannot know exactly how many worms must be right censored *at what time*. The many possibilities are bounded by two extreme scenarios.

In one extreme scenario, we hypothesize that the recaptured animals are those that went missing most recently. This scenario matches each downslope of $m(t)$ with the most recent upslope, see Fig. 8-2a. Each unmatched upslope step then provides the number of animals that were lost to the experiment at that time and must be right censored in the final assembly of the survival curve. This strategy can be formalized as a retrospective (right to left) propagation of the smallest value

of $m(t)$:

$$z(t_i) = \min(m(t_i), z(t_{i+1})), \quad i = n - 1, \dots, 1 \quad \text{with} \quad z(t_n) = m(t_n) \quad (5)$$

The series of $z(t_i)$ is a step function that increases monotonically with increasing i . A step at time t_i triggers a censoring annotation in the lifespan log, asserting that $\Delta(t_i) = z(t_{i+1}) - z(t_i)$ individuals are permanently not accounted for and should thus be right censored at time t_i .

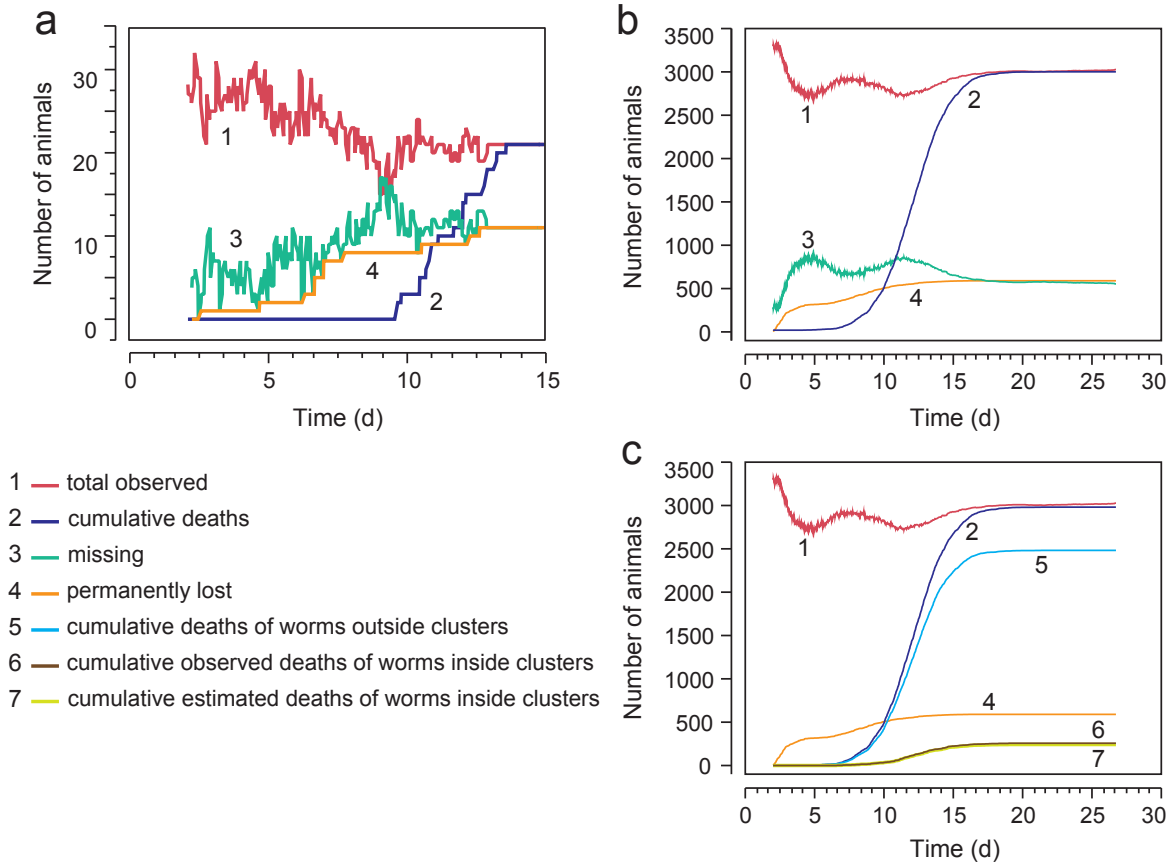


Figure 8-3: Going missing and getting lost (data). In all panels, the red curve #1 shows the total number of observed animals $obs(t)$ at time t , i.e. $obs(t) = a(t) + D(t)$ in equation (4). The green curve #3 is $m(t)$, and is obtained by subtracting the red curve from the total population size N at the beginning of the experiment. Determination of N is discussed in section 8.5. The red curve shows change at long time scales, presumably reflecting the overall clustering behavior of worms in the population. Early in the experiment, $obs(t)$ declines because a fraction of worms cluster. Worms then leave clusters, perhaps because food becomes locally scarce. As aging animals become progressively paralyzed, a subset co-locate into juxtaposed pairs or tuples. Close to death, these tuples are misidentified by our image processing as single worms, see section 8.3. The green curve $m(t)$ mirrors $obs(t)$ by construction. The orange curve #4 corresponds to the orange curve in the schematic of Fig. 8-2. **(a)** An example of missing worm detection is shown for a single plate. This plate was chosen because an unusually large fraction of animals was lost. **(b)** For each of 160 plates, the equation (5) is applied to individual plates, and the results are aggregated. **(c)** The same data in (b) is replotted with deaths separated into singletons, two-worm clusters, and three-worm clusters.

In the other scenario, starting from $t = 0$, each down-step (recapture event) at time t is randomly matched with an available up-step event (“gone missing”) that occurred before t , whereupon that up-step is no longer available for future recaptures. As in the first scenario, unpaired up-steps

correspond to lost animals (see Fig. 8-2b).

In general, we can tune the time horizon within which a recapture event is matched up with a “gone missing” event. Specifically, given a time threshold τ , we attempt to pair a down-step that occurred at time t with an up-step that occurred within $[t - \tau, t]$. If all up steps in that interval have been paired, we choose an up step within $[t - 2\tau, t - \tau]$ and so on. The first scenario corresponds to setting $\tau = 1$ (one scan interval) and the second scenario sets $\tau = T$ (the whole experiment).

In all of this, we need to know the total number of worms N at the beginning of the experiment. This is the subject of section 8.5.

8.3 Worm clusters and interval censoring

The most frequent reason for animals to remain unaccounted for in the Lifespan Machine is the irreversible clumping (clustering) of some worms near the end of their life. In the manual setting, clumping presents no problem, since the experimenter can physically separate individuals that have aggregated. There is no such intervention in the Lifespan Machine.

Early in life, when individuals are highly mobile, aggregation is dynamic and transient, yielding small clusters with odd shapes (such as a Y or T) that our image analysis software can occasionally recognize as multi-worm objects and disentangle computationally (Fig. 2-5). Later in life, when worms become stationary, clusters tend to be more persistent and worms appear to adhere to one another, acquiring an overall linear shape that our image analysis almost always misidentifies as a single worm.

Future developments in image analysis might be able to detect and disambiguate worm clusters. However, at present, the fastest and most reliable alternative is to identify and annotate clusters by visual inspection using the Worm Browser (Note 4 section 4.2). This task can be performed easily and quickly. Processing an experiment with three thousand animals takes less than an hour and is part of the data validation recommended for any novel data set to confirm that all steps of the image analysis pipeline are functioning as expected.

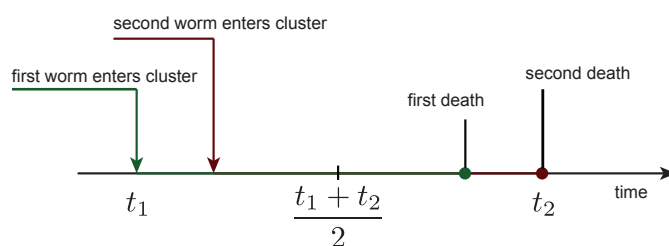


Figure 8-4: Interval censoring of a terminal worm pair (schematic). The first worm becomes stationary at t_1 . A cluster is nucleated once a second worm joins. A terminal pair is created if the latter becomes stationary in the cluster. The last of the two animals dies at t_2 (a valid individual death time). The first worm to die unnoticed within the terminal pair is then assigned (upon visual inspection of machine death calls) a death time corresponding to the midpoint of the time interval between t_1 and t_2 . To the Lifespan Machine the whole episode appears as a single worm becoming stationary at t_1 and dying at t_2 .

Although image processing typically misclassifies a persistent cluster of n worms as a single individual, the death event of a cluster is a valid event associated with the last worm dying. Once a cluster is identified, our software finds the first stationary worm that nucleated the cluster, thus defining a time interval within which the remaining $n - 1$ worms must have died, Fig. 8-4. This, then, is handled by interval censoring, either using the Turnbull estimator or the interval midpoint, see section 8.1.

8.4 Censoring strategies for worm clusters

Our current image-processing software requires manual intervention to distinguish single animals from pairs or triplets lying in close contact to each at death. To evaluate the statistical effects of such annotations, we compare 6 different strategies for handling multi-worm clusters, Fig. 8-5. All death times were acquired as discussed in Fig. 2 of the main text.

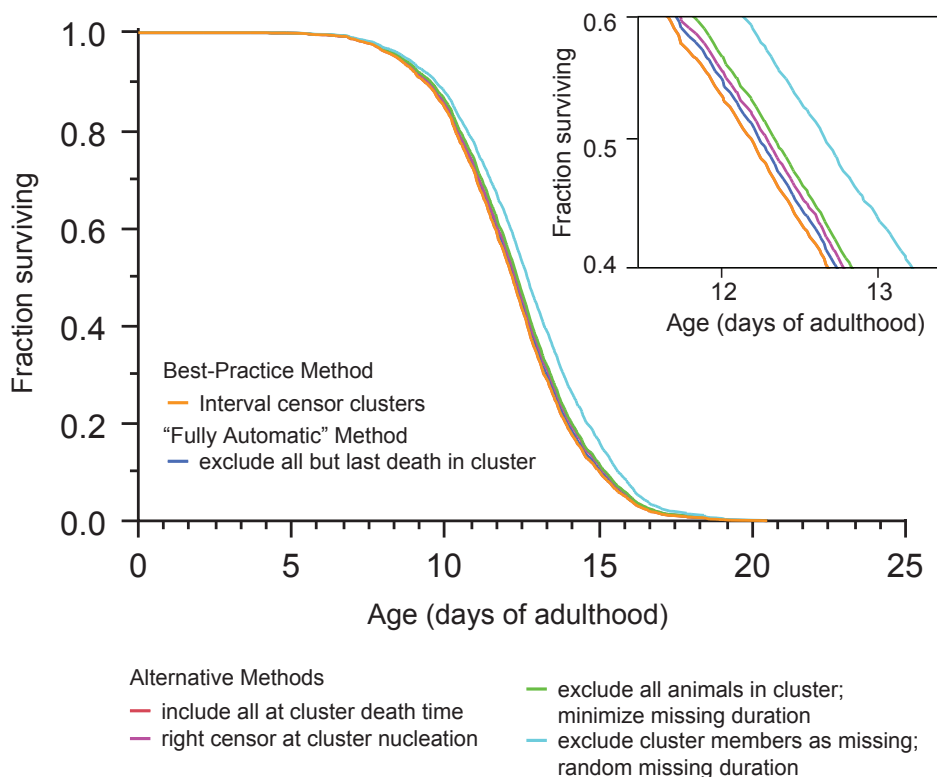


Figure 8-5: Censoring strategies for terminal worm clusters. Curve 1 (orange): Our best practice method, in which multiple-worm clusters are manually annotated using the Worm Browser. The image-processing pipeline determines an interval bounded by the nucleation time of a cluster and its death time. (The cluster is typically misidentified as a single worm.) One individual is assigned the cluster death time, all others are assigned the interval midpoint. This is the curve shown in Fig. 2b of the main text. Curve 2 (blue): No manual annotations are considered, and only a single death time is de facto recorded for all animals in a cluster. This death occurs at the death time of the last animal to cease movement within the cluster. Curve 3 (red): Using manual annotations of multi-worm clusters, all members of the cluster are assigned the same death time, which is the cluster death time. Curve 4 (purple): Using manual annotations of multi-worm clusters, worms entering into clusters are declared lost at the time of cluster nucleation and right censored. Curve 5 (green): Using manual annotations of multi-worm clusters, such clusters are completely eliminated from the death record. As a consequence, all animals that disappeared into terminal clusters are handled according to the “missing worm procedure” (which also handles transient clusters), as described in section 8.2. Curve 5 is obtained when the procedure operates according to equation (5), which minimizes the time a worm is assumed to have been missing. Curve 6 (cyan): As for curve 5, but the procedure pairs missing events and recapture events at random, see section 8.2.

8.5 Determining total population size

Sometimes the total number of worms N at $t = 0$ may be unknown; for example when animals are loaded onto plates without being counted. In this situation, we estimate the total population size by determining and comparing two quantities. First, we take the maximum of $a(t) + D(t)$ for $t = 1, \dots, T$ (see section 8.2 and Fig. 8-3). This is the largest number of worm objects ever identified during the course of the experiment. Let this number be N_0 . The visual identification of final clusters with the Worm Browser provides another estimation of the total number of worms, call it N_c . N_c is the number of worms found in clusters at the end of the experiment plus the number of singleton death events $D_s(t)$, as determined by the machine. We take N to be the maximum of N_0 and N_c .

Supplementary Note 9 – Statistical power

The effect of a perturbation on lifespan is established through comparison of the perturbed population against an unperturbed control, using a statistical test. In the case where the perturbation does have an effect, we would like the test to reject the null hypothesis that the two populations are identical not only with a sufficient level of confidence $1 - a$ (where a is the rate of false positives, type I error) but also with a sufficiently high probability $1 - b$ (where b is the rate of false negatives or type II errors). a is called the statistical significance (typically chosen as 0.05) and $1 - b$ is known as the power, typically chosen to be 0.8 or 0.9. We used our actual lifespan data to reason about the sample size required to reach a desired power.

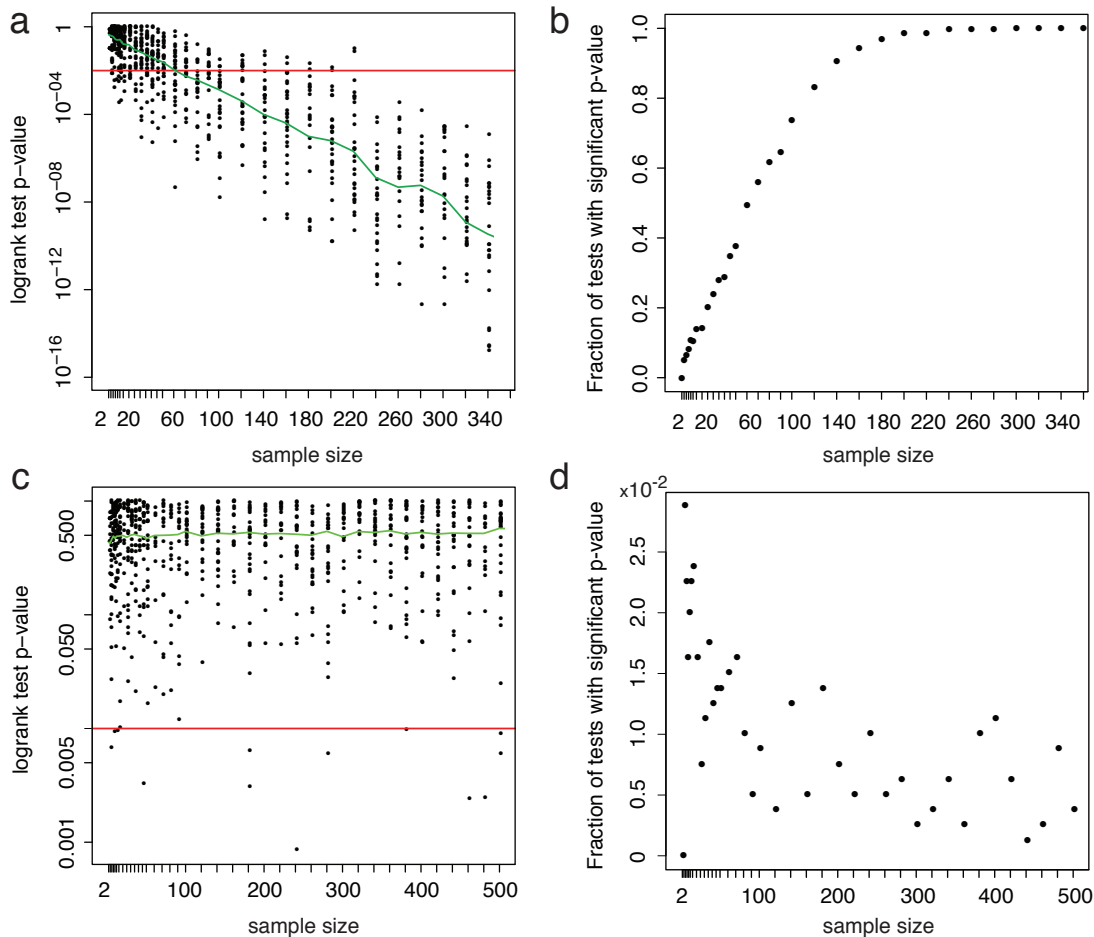


Figure 9-1: Statistical power. The top panels, **(a)** and **(b)**, illustrate the sample size dependence of the Log-rank test. In 800 replicates, we drew two sub-population of the specified size (abscissa) at random from a wildtype and *daf-16* mutant population observed at high temperature (34.5°C) by the Lifespan Machine. At this temperature the *daf-16* mutation appears to produce a 10% reduction in average lifespan (Fig. 5a). For each pair of random subsets, we used the Log-rank test to decide whether or not to reject the null hypothesis that the samples are statistically indistinguishable. Panel **(a)** shows the scatter of p-values found in each replicate at each sample size. (To allow fast rendering of the figure, only a random subset of all 800 replicates at each size are shown) Note that the ordinate of panel **(a)** is logarithmic. The red line indicates 99% confidence (i.e. 0.01 significance). Caption continued on next page.

Figure 9-1 (Cont'd): Each dot above the red line is a test with non-significant outcome, incorrectly failing to reject the null hypothesis. The green line marks the median of the p-value distributions. Panel **(b)** shows the frequency with which the (wrong) null hypothesis is correctly rejected as a function of sample size. This is known as the power of the test. (The complement of this frequency is the type II error rate or the rate of false negatives.) It can be readily seen how the power increases with sample size. The bottom panels, **(c)** and **(d)**, are controls for a case in which the null hypothesis is true—in this case the samples being compared are drawn from the same (wildtype) population (at 25 °C). Note that in both cases, the abscissa is linear in scale, but tick marks were placed at each size sampled. As for panel **(a)**, the ordinate of panel **(c)** is logarithmic. See text for details.

In one case we compare two samples, one drawn from the wildtype population and one from the *daf-16* population presented in Fig. 4b of the main text (both at 35 °C). In that case the null hypothesis is false by construction. In another case we compare two samples both drawn from the same wildtype population of 3578 individuals (25 °C) presented in Fig. 2b of the main text, illustrating the situation in which the null hypothesis is true. The red line indicates 0.01 significance (99% confidence). Each dot above the red line represents a test correctly not rejecting the true null hypothesis, whereas each dot below the red line corresponds to a test that turned out significant, thus erroneously rejecting the null hypothesis (type I error or false positive). In Fig. 9-1d we plotted the fraction of tests, as a function of sample size, that reject the true null hypothesis. As expected, this fraction is very low even at small sample sizes and approaches zero for large samples. In contrast, Fig. 9-1a reports the case in which the null hypothesis is wrong, as high-resolution data shows *daf-16* mutant populations to differ in mean lifespan from wildtype by about 10% at 34.5 °C. In the simulation, we observed sample size to have a substantial effect on power, the probability of rejecting the wrong null hypothesis at a given level of confidence, Fig. 9-1b. With a population of 110 animals every 5th test will get it wrong (a power of 80% or 20% false negatives). At around 200 animals, power has increased to 99%.

In summary, our arguments in this section indicate that sample sizes of sufficient power to detect with high confidence a difference in survival prospects of 15% or more are comfortably within reach of the manual method. Detecting 10% differences at high confidence and power becomes more tedious (on the order of 200 animals) and quickly becomes a bottleneck for determining the mortality statistics associated with a larger number of environmental or genetic conditions. The detection of small effect sizes also places more stringent requirements on the reproducibility of environmental conditions, hardly achievable with the routine manual approach. Finally, biological insights often depend on showing the *absence* of an effect, in which case very small effect sizes need to be determined with high confidence, requiring population sizes of several hundred animals. These sizes are routinely achievable with the Lifespan Machine at high throughput.

Supplementary Note 10 — Aggregation of survival curves

10.1	Device-specific environmental effects on lifespan	45
10.2	The device-effect regression model	45

10.1 Device-specific environmental effects on lifespan

The natural units of aggregation in the Lifespan Machine are its “units of confinement”: plate, scanner, incubator. The procedures described in the previous sections apply to a single plate and yield a survival curve for that plate. Each plate produces data with high temporal but low statistical resolution, as a plate contains about 30-50 worms. Better statistics can be produced by aggregating multiple plates containing animals of the same strain, either directly with our software or using a statistical package like JMP [3]. Building curves for individual plates provides a highly valuable sanity check, as it affords an opportunity for statistical tests of equivalence and for recognizing when entire plates may have to be censored (for example, due to fungal invasion, desiccation, or fogging). Data from individual plates are aggregated to generate survival curves at the level of individual scanners. Before pooling scanner-aggregated curves into an overall curve at the incubator level, they may have to be registered to account for small device-specific environmental differences, especially local temperature.

C. elegans lifespan is sensitive to environmental conditions and lifespan assays can detect the effects on survival that result from small differences in these conditions. It is therefore important to maintain a constant environment across the different plates in an experiment. This is technically difficult using the manual assay, and the problem is compounded when using a large, distributed apparatus like the Lifespan Machine. Factors such as temperature (Fig. 1-3b) or the composition of the bacterial lawn may vary slightly between plates and scanners, creating distinct micro-environments that can impose “extrinsic” lifespan variation on the “intrinsic” variation of interest. Scanners, for example, can differ in temperature because they produce and dissipate their own heat within a large confined space (incubator) whose airflow they obstruct. We see temperature differences of up to 1°C (Figs. 1-3b and 1-3c) between scanners located inside the same incubator, and identify statistically significant differences between the lifespans of animals housed on different scanners (Fig. 2c of the main text and Fig. 1-2). Since the populations on each scanner are drawn from the same homogeneous population, the differences between survival curves most likely reflect temperature differences rather than biological variation.

We deploy fans (Fig. 1-1) and optimize scanning frequencies (Figs. 1-3b) to minimize temperature differences between scanners. Ideally, these modifications would reduce environmental variation to the point where its effects fall below the detection limits of our method. This, however, is a high standard for our equipment, since the Lifespan Machine can monitor large populations at short intervals, enabling it to detect very small effects from environmental variation. We therefore pursue a different strategy: detecting and correcting such environmental variation computationally by using a linear regression model in which scanners are treated as values of a categorical variable (“device”). This model makes a few assumptions: that each scanner has the same effect on the lifespan of all individuals it measures and that it alters lifespan linearly, either shifting it by a constant offset or scaling it by a constant factor. The latter case is known as the Accelerated Failure Time (AFT) model.

10.2 The device-effect regression model

Let t_{ij} be the lifespan of the j th individual on scanner i , $i = 1, \dots, s$. The effect of environment on lifespan can be modelled in two simple ways. First, additive effects might result in time shifts between survival curves, such that t_{ij} is $\epsilon_{ij} + \beta_i$ with β_i a constant offset specific to scanner i and

ϵ_{ij} (a “residual”) the corresponding lifespan absent the “treatment” by scanner i . Second, multiplicative effects might result in a proportional scaling of lifespans, such that t_{ij} becomes $e_{ij}\beta_i$ (the AFT model). For the sake of brevity, we describe additive and proportional effects using the same (additive) model, with the understanding that in the proportional case the t_{ij} denote the *logarithm* of lifespan.

The model can be written as

$$t_{ij} = \alpha + \beta_i + \epsilon_{ij} \quad \text{if } i \neq s \quad (6)$$

$$t_{sj} = \alpha - \sum_{i=1}^{s-1} \beta_i + \epsilon_{sj} \quad (7)$$

In these equations, α is a landmark against which the device-dependent effects are determined. A possible choice is the grand mean μ of all N lifespans (or their log) measured in the same incubator or the average lifespan of a population on a particular scanner. The grand mean is a good choice when the environment varies evenly between scanners, and worms are distributed evenly across the scanners. When α is the average lifespan on a specific scanner with temperature T , we effectively correct the lifespans on all other scanners as if they were at temperature T . The $\beta_i, i = 1, \dots, s-1$ are the deviations of the mean lifespan $\mu_i = \sum_j t_{ij}/N_i$ of the N_i individuals on scanner i from α and ϵ_{ij} is the deviation of t_{ij} from μ_i . Since we must have $\sum_{i=1}^s \beta_i = 0$, β_s is determined by $\beta_s = -\sum_{i=1}^{s-1} \beta_i$.

The relation with linear regression is made explicit by introducing a categorical variable \vec{X}_j whose (vector) value encodes the scanner hosting individual j :

$$t_j = \alpha + \vec{X}_j \vec{\beta} + \epsilon_j \quad (8)$$

with $\vec{\beta}$ the vector of scanner effects to be determined. Estimates of the parameters of these types of equations are handled routinely with statistical packages such as R [4] or JMP [3].

In our case, we also must account for the fact that we place animals on scanners several days after onset of adulthood. The time that individual j is exposed to the environmental impact of a scanner is therefore not its lifespan t_j , but $t_j - t$ or $\log(t_j - t)$, where t is the age at which animals were placed on scanners.

The Cox model constitutes an additional possibility to account for a scanner effects. In Cox regression, the scanner environment is assumed to have an additive effect on the log of the hazard function. We refrain from running a Cox model, which requires fitting a baseline hazard, for the purpose of the present paper.

Any of the AFT-class of regression models can be applied to censored data using the Buckley-James estimator based on modified least-squares estimating equations [10]. This method replaces each censored observation by the conditional expectation given the observed data and the covariates. Because the conditional expectation itself depends on β , the estimating equations must be solved iteratively.

Supplementary Note 11 — Hazard

11.1	Hazard estimation and rendering	47
11.2	Hazard of wildtype	48
11.3	Hazard and sample size	51

11.1 Hazard estimation and rendering

The hazard rate is a fundamental statistical observable in aging. It is the instantaneous rate at which individuals that survived to time (age) t die in the next instant. The role played by this rate in a non-reproducing population, like a plate of sterile worms, is analogous to that of a rate constant in a first order chemical reaction that thins out an initial population of molecules by degradation. However, the hazard rate $h(t)$ is typically not a constant, but a time-dependent quantity that may (and often does) increase with time reflecting the functional decline associated with aging: $ds(t)/dt = -h(t)s(t)$, where $s(t)$ is the survival function. $h(t)$ can be observed directly by counting the frequency of death events relative to the survivors (the population at risk) at the beginning t of intervals of fixed duration Δt (“period hazard”): $h(t) = [s(t) - s(t + \Delta t)]/\Delta t s(t)$. To do this accurately, however, requires population sizes that are larger and more frequently observed than is typically the case in manual lifespan experiments. The Lifespan Machine makes this fairly straightforward, and the machine can be scaled up, if desired, to test many conditions.

We determined three kinds of parametric hazard functions for our data: Gompertz, Weibull and Log-logistic. Gompertz and Weibull hazards represent exponential and polynomial increases in the risk of failure, respectively, and are widely used to describe mortality data in biological and technological systems. The Log-logistic is a Weibull distribution with a deceleration term. We used it here as a reference to compare the deceleration of mortality rates observed in most cases. Such deceleration has been described anecdotally before in *C. elegans* [20–22] and flies [23, 24] (and humans, wasps and automobiles [20]), although never in stress regimes (e.g. high temperature) nor systematically for a large number of mutants.

The parameters of model hazard functions—Gompertz, Weibull, and Log-logistic—were obtained by maximum likelihood estimation (MLE) using the lifespan data (death times) of the wildtype population shown in Fig. 2 of the main text. Gompertz and Weibull hazards do not decelerate and MLE estimates of the parameters of these distributions will yield a poor fit. The origins of the deceleration phenomenon deserve an empirical study in their own right, which we leave to future work (but see [25–27] for computational and theoretical approaches). In the present data analysis we focus on the functional form of early mortality, which is a natural and easier first task in organizing our volume of data. To this end—and unless otherwise noted—when obtaining MLE estimates of model parameters, we only use lifespan data up to the median, right-censoring the remainder and thus eliminating the late phase in which hazard decelerates. This is what we mean in legends or captions when referring to, for example, “50% Gompertz”. The phrase “100% Weibull” then means that the full range of lifespan data was used, except for cropping 1% at the noisy tails. The maximum likelihood estimation of model parameters was performed using the `flexsurv` package of R [4].

A quantile-quantile (Q-Q) plot is a diagnostic for assessing the extent to which survival data conform with the distributions whose parameters we determined. In a Q-Q plot we graph the time it takes a functional form (Gompertz, Weibull, or Log-logistic) to reach a certain quantile against the time it takes the empirical survival function to reach the same value. In our figures we place the former on the abscissa and the latter on the ordinate. The values (plotting positions) are determined by the steps in the Kaplan-Meier survival curve obtained from our data (Note 8 section 8.1). The 95% confidence intervals in the Q-Q plot derive from the confidence intervals generated

in the survival curve assembly from data (Greenwood's formula) using the `survfit` function of R's `survival` package. The extent to which empirical data and (MLE) fitted functions coincide can be assessed by how closely the points hug the identity line. It is worth pointing out that while we determine parameters based on the data up to the median lifespan, we extend the Q-Q plot to include the entire range of data. This makes the onset of deceleration neatly visible as an upward departure from the diagonal for Gompertz and Weibull hazard functions.

11.2 Hazard of wildtype

The next two figures evidence that high-resolution lifespan data for wildtype conform with a Weibull distribution better than a Gompertz distribution up to times slightly beyond median lifespan when deceleration becomes noticeable.

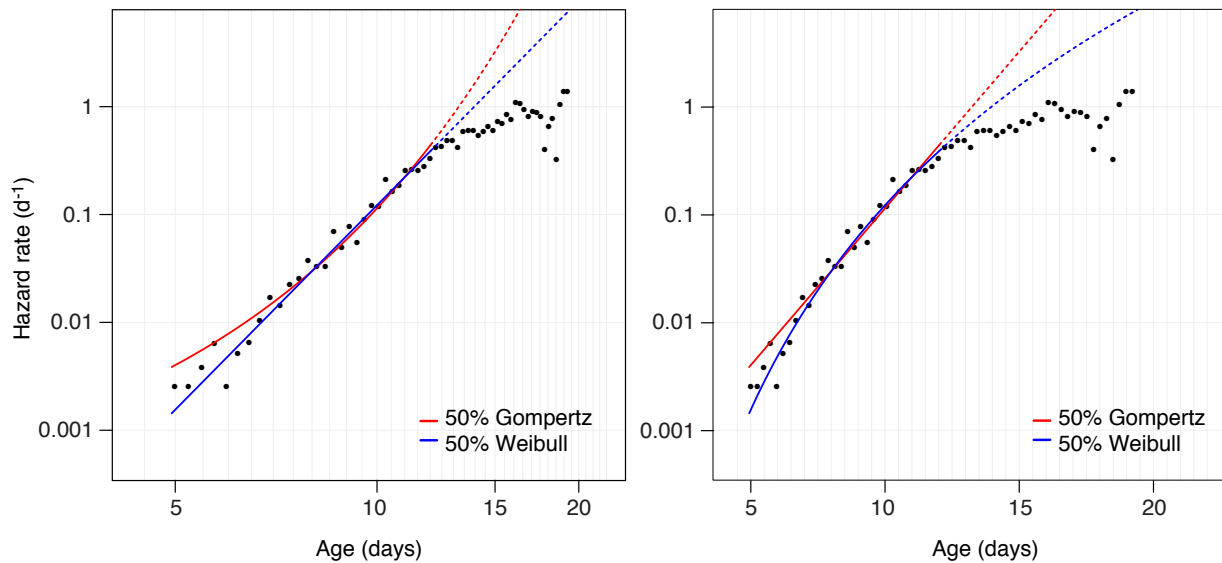


Figure 11-1: Wildtype high-resolution hazard functions at 25 °C. In this figure we graph, for the purpose of visual comparison, our hazard rate data (black dots) and the Gompertz (red) and Weibull (blue) model hazard functions as determined by MLE from our lifespan data. The black dots are produced by subdividing the duration of the experiment into equal intervals and reporting the average risk of death (the number of deaths as a proportion of the population initially at risk) within each interval. The comparison with model hazard functions suggests that a power-law (Weibull) describes early mortality better than an exponential (Gompertz). The left panel reproduces Fig. 2d of the main text and is plotted using a log-log scale (in which the Weibull hazard is a straight line), while the right panel uses a log-lin scale (in which the Gompertz hazard is a straight line). The hazard fits are drawn solid over the data range used in the fits (i.e. up to median lifespan) and dotted elsewhere.

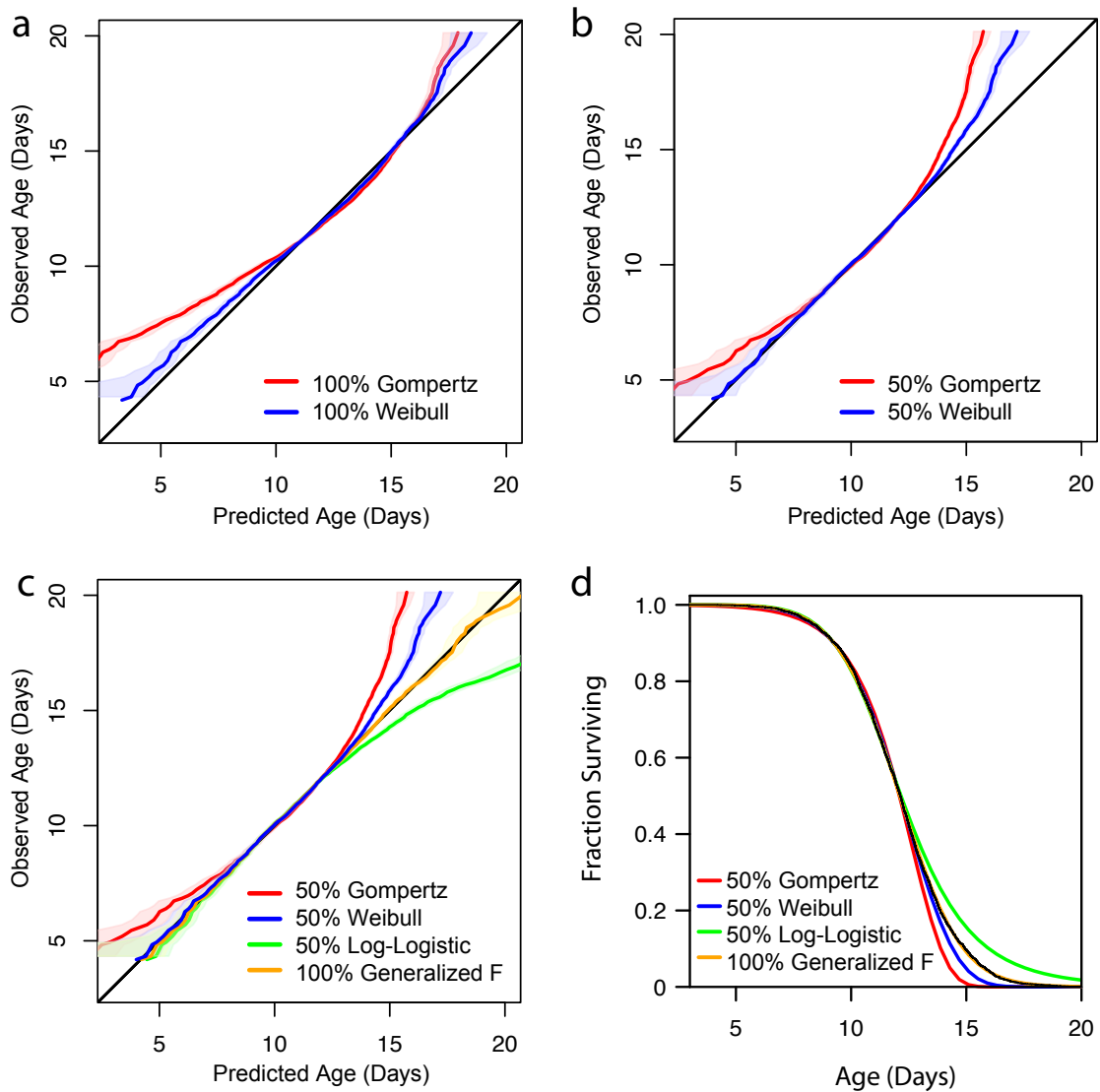


Figure 11-2: Wildtype high-resolution quantile-quantile plots at 25 °C. The quantile-quantile plots (see section 11.1) visualize the extent of (dis)agreement between data and a variety of models. **(a):** The parameters of Gompertz and Weibull hazards were determined by MLE using the full range of data (except for the 1% crops at the tails). Clearly, the agreement is poor, since the hazard deceleration present in the data is at odds with functions that cannot decelerate. **(b):** Model parameters were determined using data only up to the median lifespan. This panel is a striking indication that early mortality in wildtype *C. elegans* follows a Weibull more closely than a Gompertz. **(c):** Two further models are considered: a Log-logistic, whose parameters were determined using data up to the median, and a generalized F distribution [28] that takes into account the whole range of data. The Log-logistic coincides with the Weibull distribution at early times, but evidently overestimates the extent of hazard deceleration. The generalized F is a four-parameter umbrella that encompasses both Log-logistic and Weibull distributions. Not surprisingly, it can be tailored to the data very well. Panel **(d)** shows the survival curves implied by the fitted model parameters. The black curve with the narrow 95% confidence interval represents the non-parametric Kaplan-Meier curve based on our data. The generalized F sits snug underneath. The main differences to the other models are chiefly in the tail. Again, the blue Weibull curve is seen to match the data well beyond the median.

The hazard analysis of the manually scored results also indicates a better fit to the Weibull than the Gompertz distribution.

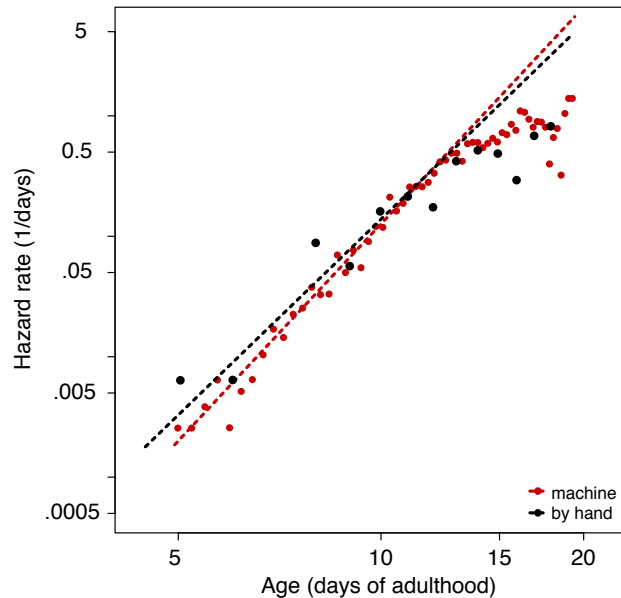


Figure 11-3: Wildtype hazard at 25 °C scored manually. This figure compares the empirical hazard of the manually scored wildtype population (Fig. 2 in the main text) with that of the LM-scored population. The Weibull hazard is seen as the better fit for early mortality data obtained with either method.

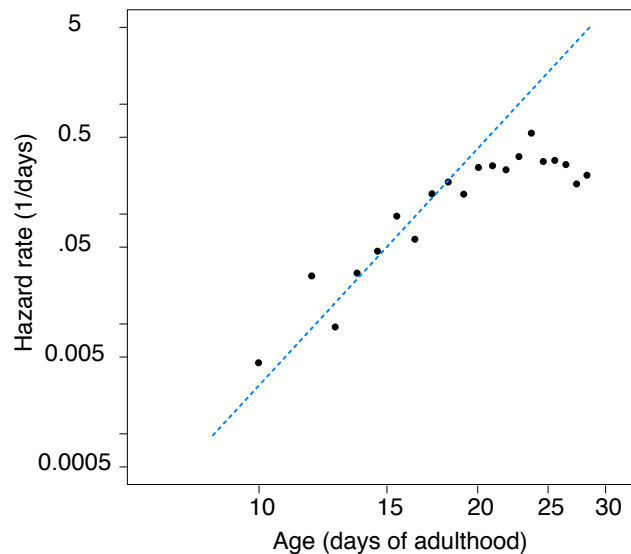


Figure 11-4: Wildtype hazard at 20 °C on HT115. This figure depicts a hazard rate plot with Weibull fit obtained from a wildtype population (220 animals) feeding on HT115 bacteria at 20 °C. It shows that the general features of the mortality kinetics at 25 °C persist at 20 °C.

11.3 Hazard and sample size

In Fig. 11-5 we demonstrate the impact of population size on hazard estimates. The effects of population size demonstrates the precision afforded by large populations.

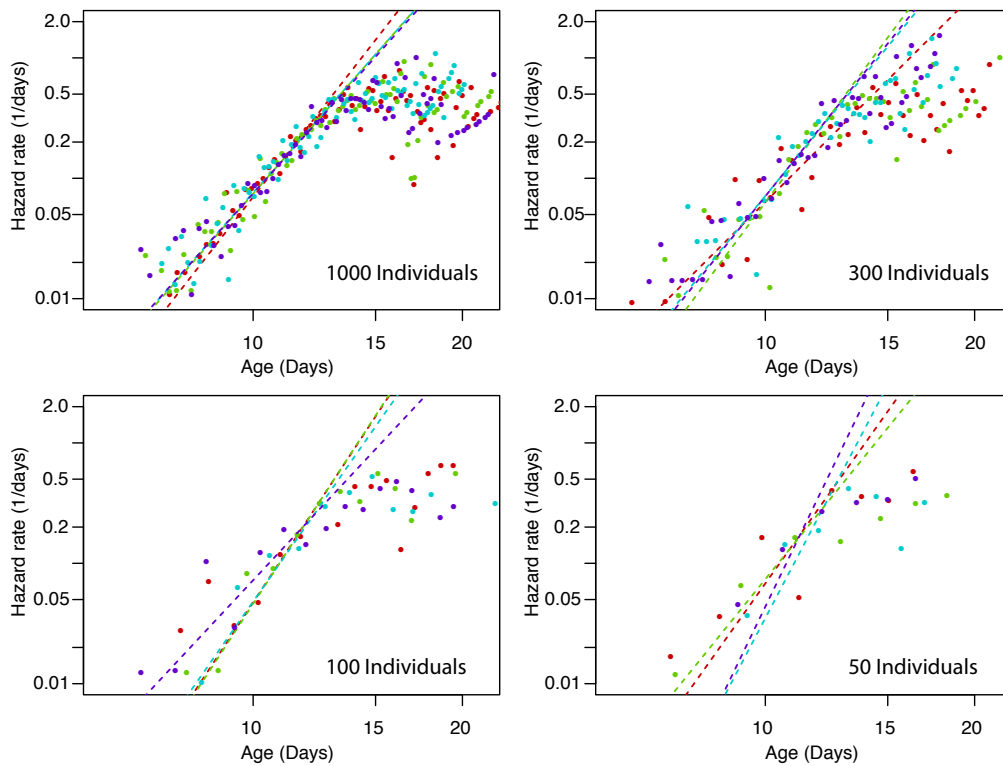


Figure 11-5: Population size and hazard parameter estimates. The four hazard panels show empirical hazard data (dots) and Weibull fits (MLE parameter estimates to original lifespan data) for four random samples (colors) at the indicated sample size. This conveys a sense of how sample size impacts the estimation of hazard parameters.

Supplementary Note 12 – Control for scanner effect

C. elegans researchers define death in terms of macroscopically observable proxies. The most widely used proxy is the failure to move in response to prodding with a platinum wire. Our automated approach uses a slightly different proxy—the permanent cessation of all spontaneous motion as determined by retrospective analysis of long-term observations. We observe these two proxies correlate surprisingly well (Fig. 2b and Fig. 12-1). Yet, absent a principled theory of death, it seems conceptually impossible to “control” for one proxy by comparing it to another. Specifically, we cannot expect the comparison of a manually performed lifespan experiment to act as a strict control for an automated experiment, as the two measure somewhat different phenotypes. This is why we established quality and consistency of our method in two ways: First, by evaluating the Lifespan Machine’s ability to recapitulate in a self-consistent fashion known effects on lifespan resulting from a wide variety of interventions, such as mutations, RNAi, and exposure to stressors of physical (heat) and chemical (t-BuOOH) nature (Figs. 3 and 4 of the main text). Second, by developing software facilitating the human *ex-post* inspection and validation of the visual record associated with an experiment.

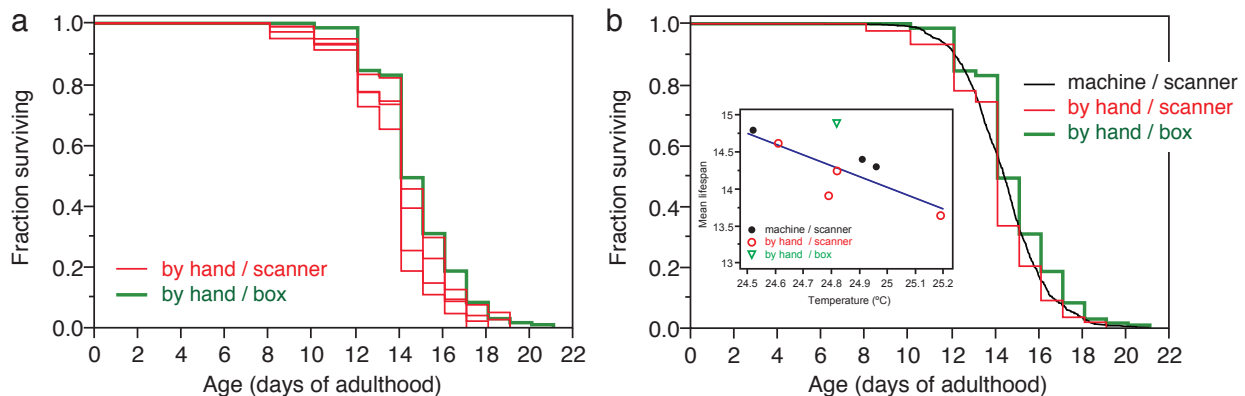


Figure 12-1: Absence of a “scanner effect”. To test whether the scanner environment affects the lifespan statistics of worms, we manually scored plates residing on scanners, thus exposed to the same environment as plates processed by the Lifespan Machine. Panel (a) compares a set of such manually acquired survival curves with curves assembled from the traditional manual scoring of plates stacked in a box inside the same incubator. Both curves agree. Panel (b) compares the two manual approaches (plates inside a scanner or in a box) with the Lifespan Machine. The inset of panel (b) shows the temperature differences between animals on individual scanners, monitored by various techniques, inside the incubator set at 25 °C (see also Fig. 1-2). All plates were seeded with HT115(DE3), since the data were acquired in conjunction with Fig. 13-3.

We sought to determine what effects the scanner environment (light, vibrations, etc.) might have on nematode lifespan. For example, a stimulation of worm movement in response to light has been noted using standard microscopy techniques [11]. Mathew et al. [12] suggest that the light produced by flatbed scanners may elicit a similar response. We transferred several thousand wildtype animals onto plates stored in two environments—inside operating flatbed scanners and inside a box sitting inside a neighboring, temperature-matched incubator. For animals stored inside flatbed scanners, we observed a portion by hand and the remainder using the Lifespan Machine, Fig. 12-1.

We first discuss animals scored by hand. We observed that animals located inside scanners lived 18 hours shorter than those housed outside scanners in a separate incubator. To place this difference in the context of temperature, we identified a significant correlation between measured scanner

surface temperature and mean lifespan, both through linear regression on plate means ($R^2 = 0.65$; $p = 0.027$) and in Cox regression with temperature as a continuous covariate ($p = 0.002$). Using our linear regression as a predictor for lifespan we found that the difference between scanner and box groups was reduced to 15.6 hours, or 4.4% of lifespan, representing our best estimate of the effect of the scanner micro-environment on nematode lifespan. We suspect that effects of this magnitude, despite our attempts at accounting for the effect of temperature, might nevertheless result entirely from limitations in our ability to accurately measure temperature as it varies between scanner and box environments.

We then considered animals housed on scanners, scored either by hand or the Lifespan Machine. We found that animals observed by hand appeared to live 6.2 hours shorter than those observed by machine, or 1.8% of total lifespan. This close correspondence between our method and the manual procedure suggests that our proxy for death, cessation of movement closely matches that of the manual, and that any worm that is capable of spontaneous movement does eventually move.

Supplementary Note 13 – Additional survival curves, controls

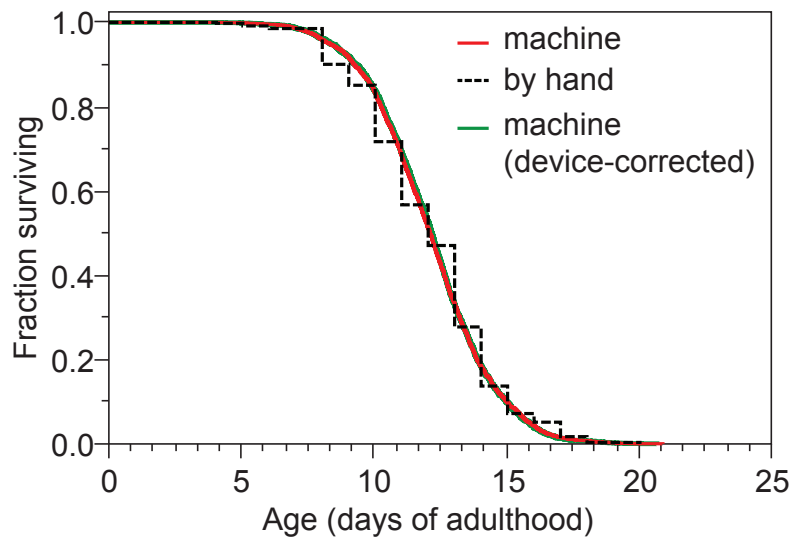


Figure 13-1: Wildtype at 25°, device-corrected. In Fig. 2 of the main text, panel 2b shows the wildtype survival data aggregated from the entire experiment and panel 2c shows survival data from individual scanners with their device corrected versions using the categorical regression of Note 10 section 10.2. This figure shows, for completeness sake, the aggregated device-corrected curve alongside the curves of panel 3b for comparison. At the aggregate level, device correction is seen to have only a minor impact in this case, suggesting that the small variations in temperature across scanners and their effects nearly cancel out.

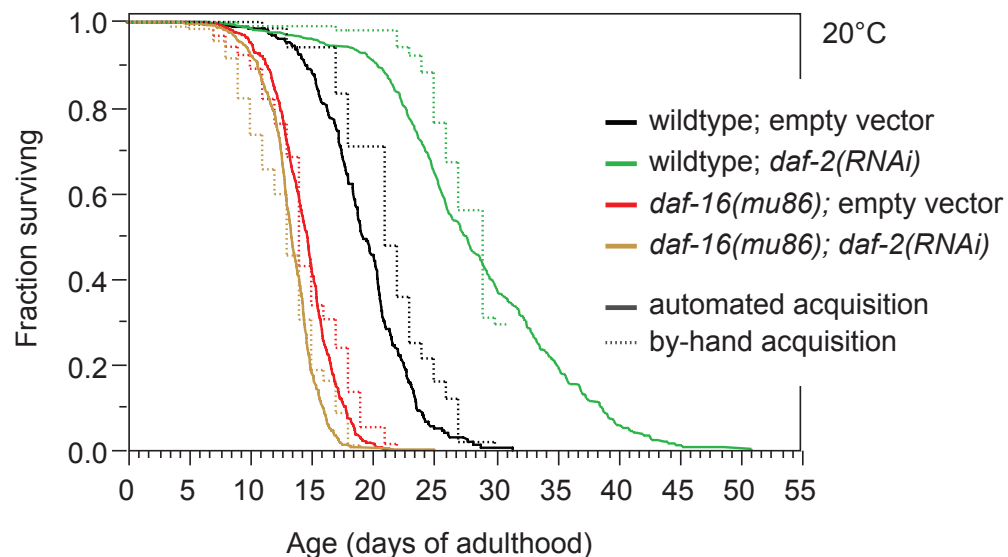


Figure 13-2: RNAi Lifespans at 20 °C. To test the performance of the Lifespan Machine at 20 °C, we placed a subset of animals prepared for Fig. 3a in the main text on scanners in an incubator calibrated at 20 °C. The by-hand control for wildtype on *daf-2(RNAi)* was truncated as experimenters left for winter holidays; the Lifespan Machine continued unaffected.

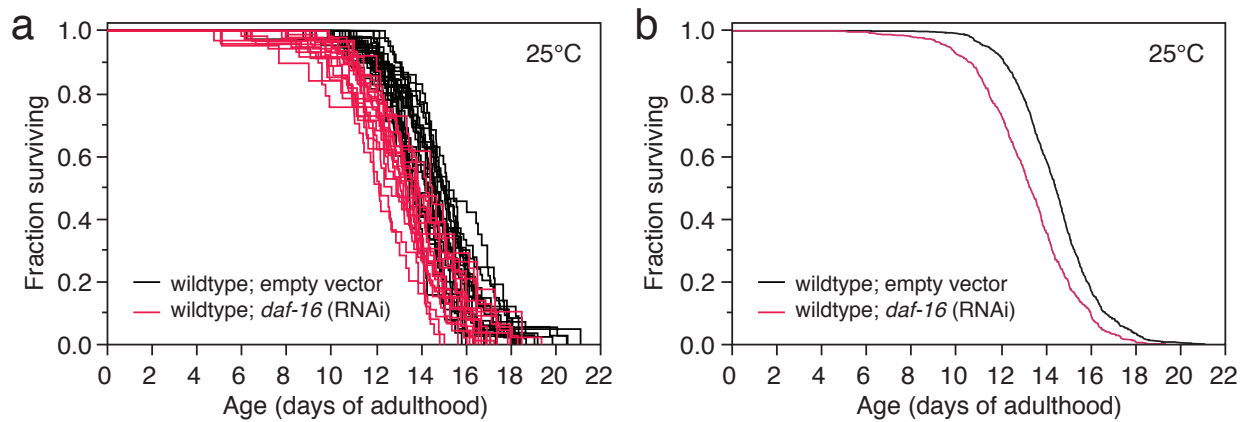


Figure 13-3: *daf-16(RNAi)* at 25 °C. RNAi of *daf-16* has been reported to shorten lifespan [13,14]. Here we note that the RNAi knockdown is noticeably weaker than the mutant shown in Fig. 3c of the main text. Panel **(a)**: survival curves by plate. Panel **(b)**: aggregated survival curves.

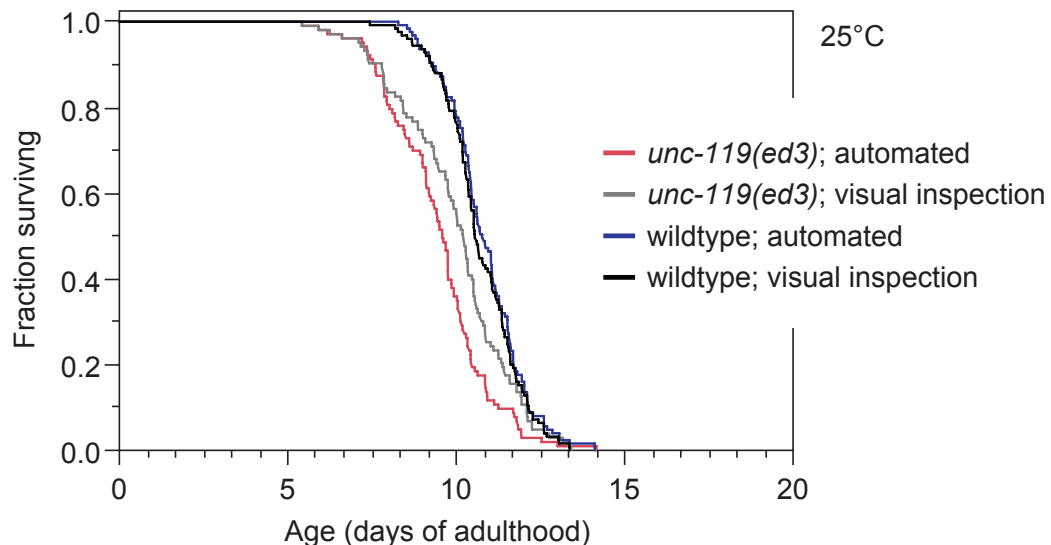


Figure 13-4: Automated lifespan assay of a severely uncoordinated mutant. We used *unc-119(ed3)* animals to assess the performance of the Lifespan Machine on worms with unusual movement phenotype. These animals show abnormal development of the nervous system and exhibit both slow and uncoordinated movement [15]. Using the posture analysis parameters optimized for wildtype animals, our image analysis consistently identifies death times of *unc-119* as being, on average, only 1.5 days shorter than wildtype. The death times of *unc-119* were slightly underestimated by 0.3 days in mean lifespan compared to the visual inspection. This difference is not statistically significant ($p = .09$). Movement parameters can be optimized for specific behavioral mutants, if necessary.

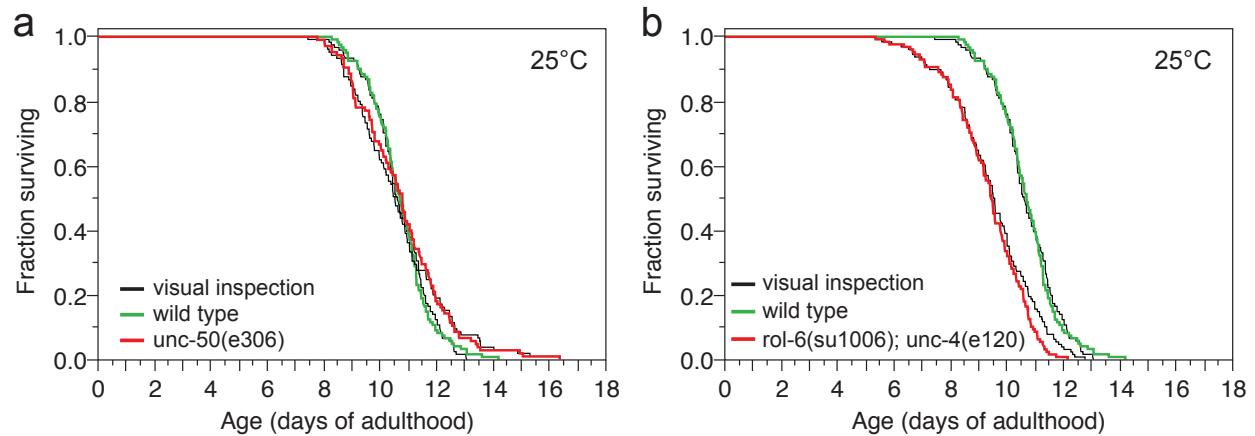


Figure 13-5: Survival curves of *unc-50(e306)* and *unc-4(e120);rol-6(su1006)* at 25 °C. Panel (a): Hermaphrodites possessing the *unc-4(e120)* allele have been previously characterized [16] as having a significantly shorter lifespan than animals with the wildtype allele. Pairwise comparisons between automated analysis (colored curves) and visual inspection of the image record (black curves) are provided as in Fig. 3d of the main text. Panel (b): The inclusion of a strain exhibiting a roller phenotype further demonstrates the broad applicability of our approach.

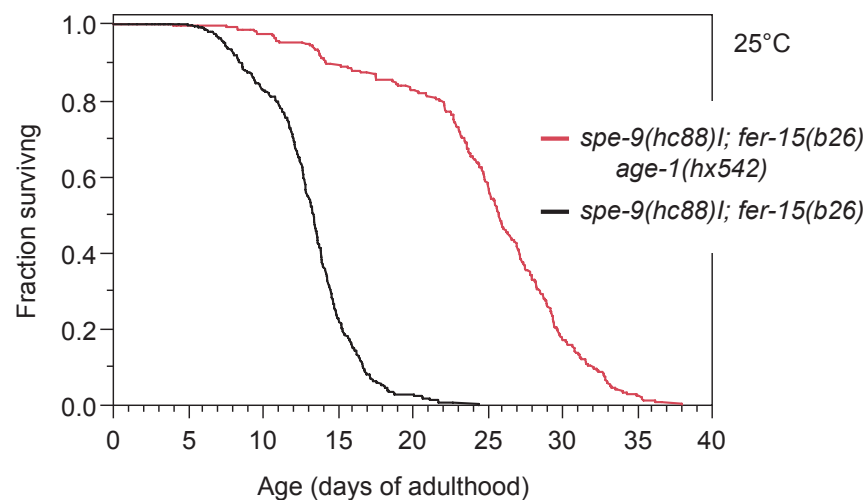


Figure 13-6: Lifespans of temperature-sensitive self-sterile mutants. To avoid the potential for cohort confusion, lifespan experiments with self-fertilizing hermaphrodites on the same plate must prevent the accumulation of live progeny. This is often achieved with 5-fluoro-2'-deoxyuridine (FUdR) [17]. To assess the performance of the Lifespan Machine without the use of FUdR, we assayed two strains containing temperature-dependent sterile mutations in the *spe-9* and *fer-15* genes. Placing the eggs of such mutants at 25 °C results in sterile populations.

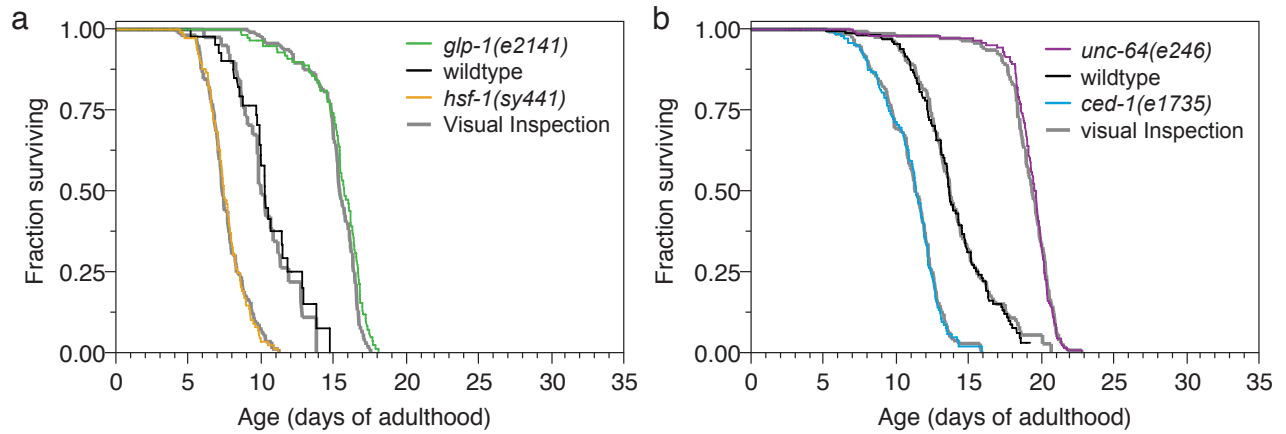


Figure 13-7: Visual controls of additional mutant survival curves. Fig. 3 in the main text misses the juxtaposition of the survival curves acquired by machine with those obtained by visual inspection with the Worm Browser for the mutants shown here in panels **a** and **b**. Visual and automated curves were indistinguishable by Log-rank ($p > 0.04$). *glp-1* and *hsf-1* strains were grown at 15°C before being shifted to 25°C.

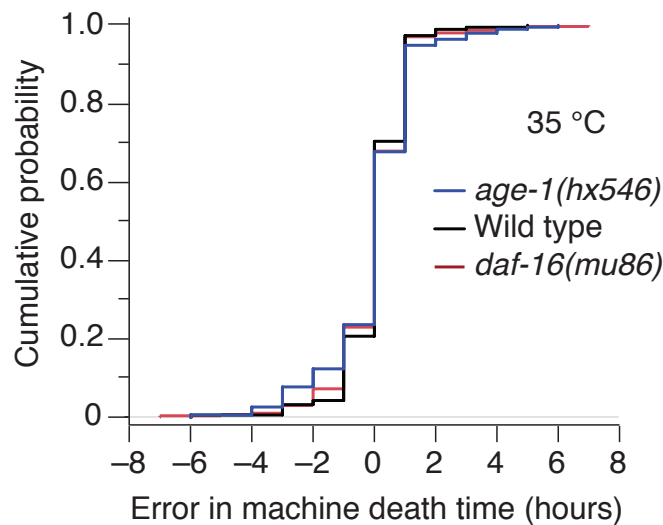


Figure 13-8: Error of LM operation at high-temperature. To complement the visual validation of the LM at 35 °C, we plot the LM error (as in Fig. 2f of the main text), associated with Fig. 4c of the main text.

Supplementary Note 14 — Limitations

As a system, the Lifespan Machine is best evaluated in terms of the quality of the survival statistics it produces. Yet, components such as imaging hardware, worm classifiers, and movement detection algorithms, can also be evaluated in isolation. For example, adoption of imaging devices other than flatbed scanners (e.g. cell phone cameras) may decrease both the physical bulk of the Lifespan Machine and the potential for temperature variation between plates. Furthermore, we find that our SVM classifiers produce the best results on younger worms, strains with larger bodies, and plates with thinner agar lawns, as these situations yield more distinct images of worms. Our posture analysis algorithm works best on animals that do not exhibit long pauses in late-life movement, as our use of a pause threshold to remove noise works less well in these cases. Occasional imperfect performance in all these components can be seen by scrutinizing Supplementary Videos 1–5. We expect that improvements will be made in many components of the Lifespan Machine. For example, movement time-series analysis might use Hidden-Markov models, or worm detection strategies might integrate image data from consecutive frames. The consequences of such local improvements, however, should be evaluated with regard to both the independent behavior of the component and its integrated effect on the overall quality of the survival curve.

Beyond transient limitations that are more reflective of the current state of development of the Lifespan Machine, our approach is restricted in some more fundamental aspects.

- *Worm density on a single plate*

Our software can analyze 100 or more worms on each plate. In practice, however, worm behavior introduces complexities that limit the usefulness of our analysis of high-density populations. At all densities, we see animals often die closely as pairs, juxtaposed head-to-head and tail-to-tail. This circumstance makes it hard for the image analysis to *automatically* distinguish a single dead animal from a pair of dead animals stuck together (Note 8 section 8.3) These cases we resolve through rapid visual inspection using the Worm Browser (Note 4 section 4.2). For many users, it may be permissible to instruct the survival curve assembly software to discard such pairs entirely, as doing so does not appear to cause problems of misestimation, see Fig. 8-5, presumably because such “paired dying” occurs randomly. At high densities, however, we see an increasing number of animals dying in groups of four or more, at which point even careful human inspection cannot disentangle the various death times. Ultimately, we prefer to minimize the potential scope of such problems by maintaining low animal densities. We find that with 35 animals per plate the frequency of worms dying in close juxtaposition is very tolerable. Lowering worm density can further reduce issues associated with worm aggregation, if desired.

- *Deaths from causes unrelated to aging*

The current version of the Lifespan Machine cannot discern causes of death that are unrelated to aging, such as bagging. The risk of bagging (and perhaps of non-aging deaths in general) decreases as egg production slows with age. As a result, bagging occurs almost exclusively during a nematode’s reproductive window. In our by-hand data sets the frequency of such events is less than 5% and in some cases 0%, agreeing with previous characterization [6]. Bagging and vulval ruptures can be identified in the visual data validation step using the Worm Browser, where they can be annotated for censoring. In Fig. 8-5, we demonstrate how various strategies for handling a much more frequent censoring event—the disappearance of worms—have measurable but ultimately negligible effects on survival curves. We therefore do not expect typical bagging rates to have much of an impact on the quality of survival data produced by the Lifespan Machine. However, certain environments and specific mutations may increase the rate of bagging, rendering automated analysis infeasible. In such cases we

recommend the following strategy for recognizing (and removing, if desired) bagged animals. Because bagging occurs primarily early in adulthood when deaths are very infrequent, animals need not be placed onto scanners until late in their reproductive span. Populations can be manually inspected for bagging (or any other visible phenotype) for several days of adulthood, after which they are transferred into the automated imaging apparatus for survival analysis.

- *Inaccessibility of plates and animals during an experiment*

A clear limitation of our approach is that animals cannot be handled during automated observation. It follows that nematodes must be sterile during observation, so that progeny do not obscure the analysis of parents. One common means for ensuring sterility is to add FUdR to media. If FUdR is problematic (affecting, for example, the robustness of specific RNAi effects), there are several options available to the experimenter, such as the use self-sterile mutants (see Fig. 13-6). If the experimenter can tolerate some loss of precision during the first few days of death, animals may be passaged by hand as young adults and then loaded onto scanners after they cease laying eggs during mid-life (as in the bagging case described above). If the research community as a whole decides to move away from FUdR, new methods will need to be devised to prevent live progeny. Because of our adherence to standard culturing techniques, these methods are likely to be applicable in the context of the Lifespan Machine as well. Another is the use of temperature-sensitive sterile strains. However, these solutions may not always be appropriate, and in certain cases the reproductive span of individuals may persist into late ages, precluding the simple solution of transferring animals onto scanner only after they have ceased reproducing. The Lifespan Machine would not be very useful in such instances.

In the main text we demonstrate the use of the Lifespan Machine as an assay for survival under exposure to the oxidant t-BuOOH (Fig. 4 of main). Quite generally, our method can be employed to assess the impact of chemicals on lifespan. The inaccessibility of plates, however, requires that chemicals must persist at defined initial concentrations throughout an experiment, as they cannot be replenished.

- *Traits that may affect the acquisition and interpretation of lifespan data*

There may be mutants whose behaviors makes automated analysis impractical; for example mutants that show no movement late in life, or mutants that crawl off the plates at a high frequency, or mutants that bag cryptically late in life. The possibility of these cases underscores the importance of quality control with the Worm Browser to detect them.

Supplementary Note 15 — Resolution and scalability

The apparent smoothness of a survival curve depends not only on population size but also on sampling (scanning) frequency. A large population that is observed infrequently will yield a survival curve with large steps, as many deaths accrue between successive measurements. An example is the wildtype manual control presented in Fig. 2b of the main text, where a population of ≈ 500 animals is observed at a frequency of once a day, producing large steps near the mean lifespan. At the other extreme, frequent observation of small populations also yields survival curves with pronounced steps, as in that case each individual death accounts for a sizeable fraction of the population. Evidently, some advantages of large populations are only fully realized when paired with frequent measurements. Therein lies one advantage of an automated method like the Lifespan Machine. Another advantage is scalability. When the overall lifespan of a population is short—as it is, for example, at 35°C (Fig. 4 of the main text)—even our baseline observation frequency of 24 measurements per day permits relatively few observations, potentially eliminating the advantage of large population sizes. In such a case, however, we can increase the effective temporal resolution by using multiple scanners and staggering their scan times to effectively monitor at all times some sub-population of the total population residing in the apparatus. We use this technique to improve the temporal resolution of 35 °C survival assays, resulting in smoother aggregated survival curves (Fig. 4b of main).

Supplementary Note 16 — Use cases

The lifespan machine captures images of worms using flatbed scanners, whose data are analyzed and integrated by software. Scanners can be easily added and removed, allowing the technique to be applied at small scale in installations including one or two scanners (as in Fig. 3c, e, f of the main text, where four mutants were monitored in two rounds on two scanners), or at large scale (as in Fig. 2b, where one mutant was monitored across ten scanners) As such, the method reduces labor in a variety of experimental contexts; adding scientific value by enabling new or old (and recalcitrant) questions to be approached with commensurate data.

- *One or two scanners*

A single scanner holds 16 plates with 35 worms each for a total of 560 individuals. As indicated in Note 9, this capacity is sufficient to detect 10% lifespan differences between two genotypes at 99% confidence and 99% power. Therefore, a single scanner enables standardized, high-resolution survival assays, appropriate for detecting subtle genetic effects. A single scanner installation involves three components—an incubator (for temperature control), a scanner (for image capture), and a computer (to control the scanner and store data). The incubator is standard equipment in most labs working with *C. elegans*, and small installations may simply use one shelf of a pre-existing incubator.

The use of single-scanner installations constrains the number of different populations that can be simultaneously monitored and the statistical power at which those populations can be characterized, limiting the number of possible applications. For example, the characterization of mortality-rate deceleration may not be feasible using a single scanner, since hazard deceleration becomes prominent only after the median lifespan (at which point the dwindling population size attainable with single scanners may provide insufficient statistical power). Moreover, we find several instances where the proportional hazards assumption of certain statistical tests (e.g. Fig. 3b) are not appropriate, necessitating the use of less powerful statistical approaches requiring larger populations to be monitored. All of these limitations are compounded by the loss of worms (Note 8 section 8.1) from plates, as well as the loss of entire plates due to fogging or desiccation, which often lowers the population sizes captured on a single scanner from their theoretical maximum by about 10

- *Five or Ten Scanners*

The limitations of single-scanner installations are easily alleviated by using multiple scanners. The lifespan machine is designed to allow multiple scanners to be deployed with little additional difficulty or complexity, compared to setting up a single scanner. If a laboratory acquires a dedicated computer and incubator for automated lifespan analysis, it is not very laborious to fill the incubator with scanners. A large incubator with 5 shelves will hold 10 scanners, forming a natural “medium” sized installation. Scaling up from a single scanner to ten scanners is merely a matter of purchasing, modifying, and plugging in the additional scanners; our image server software is designed to accept new scanners even while experiments are running.

A ten-scanner installation affords population sizes that provide greater statistical power not only for standard analyses such as the Log-rank test or Cox regression, but also for the application of somewhat more exotic analyses, such as non-proportional-hazards approaches or techniques that directly estimate the time-dependent hazard function. Our intended purpose of providing high statistical power is not to simply permit detection of increasingly small lifespan effects, as there will be a point when statistical significance ceases to correlate with biological significance. Instead, the machine is meant to enable investigations into the shape

and functional form of the hazard rate, which often requires the evaluation of subtle effects that occur late in life, at which time effective population numbers have dwindled, as well as early in life where only a fraction of deaths in each population occur. Installations of five to ten scanners allow the experimenter to perform multiple replicates of these types of analyses.

Installations of five to ten scanners also permit experimenters to screen through libraries of mutants and RNAi constructs. When population sizes of 100 animals are chosen, three strains can be reasonably tested per scanner, allowing thirty strains to be measured in a single incubator. Consider that a wildtype population should be run on every scanner along with the target populations. This will necessarily curtail real estate for mutants that can be co-located on the same scanner. One should also consider that this use captures entire survival curves for each population, increasing the quality but necessarily decreasing the throughput compared to the single time-point assays currently used in many high-throughput screening contexts.

Stress resistance assays, such as the thermotolerance experiments shown in Fig. 4 of the main text, also benefit from an incubator full of scanners. Being able to spread animals across multiple scanners allows observation times to be staggered, thereby permitting a finer temporal resolution for capturing the time-dependent hazard function.

- *Twenty or more scanners*

Construction of a scanner farm requires the dedicated use of several incubators or a temperature controlled room. At this scale, any number of projects becomes possible, including further development and testing of the apparatus itself in parallel with running experiments. The tunable statistical resolution makes our method ideally suited for large rapid reverse-genetic and chemical screens for effects on survival at intermediate or higher statistical resolution.

Acknowledgments

We would like to thank Joy Alcedo for providing the *hsf-1* and *glp-1* mutant strains, Becky Ward and Debora Marks for critical reading of our manuscript and Catalina Romero, Debora Marks and members of the Fontana lab for helpful discussions and encouragement throughout this project. We would like to thank Tom Kolokotronis, Eric Smith, and Lee Jen Wei for discussions and statistical advice, and Mason Miranda our departmental IT specialist, for patiently meeting our unbounded needs for data storage. This work was funded by NIH grants R03 AG032481, R03 AG032481-S1, and R01 AG034994-01.

References

- [1] Jones J (2008) GIMP User's Manual.
- [2] Boser BE, Guyon I, Vapnik V (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM Press, pp. 144–152.
- [3] Sall J, Lehman A, Stephens M (2012) JMP Start Statistics: a Guide to Statistics and Data Analysis Using Jmp, Fifth Edition. SAS Institute.
- [4] R Development Core Team (2009) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>. ISBN 3-900051-07-0.

- [5] Kuhn H (1955) The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 53.
- [6] Apfeld J, O'Connor G, McDonagh T, DiStefano PS, R C (2004) The AMP-activated protein kinase AAK-2 links energy levels and insulin-like signals to lifespan in *C. elegans*. *Genes Dev* 18: 3004–3009.
- [7] Kaplan EL, Meier P (1958) Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association* : 457–481.
- [8] Turnbull BW (1976) The empirical distribution function with arbitrarily grouped, censored and truncated data. *Journal of the Royal Statistical Society Series B* 38: 290–295.
- [9] Stata Corporation (2007) *Stata Base Reference Manual* : Release 10. College Station, TX : Stata Press.
- [10] Buckley J, James I (1979) Linear regression with censored data. *Biometrika* 66: 429–436.
- [11] Ward A, Liu J, Feng Z, Xu XZS (2008) Light-sensitive neurons and channels mediate phototaxis in *C. elegans*. *Nature Neuroscience* 11: 916–922.
- [12] Mathew M, Mathew N, Ebert P (2012) WormScan: A Technique for High-Throughput Phenotypic Analysis of *Caenorhabditis elegans*. *PLoS ONE* 7: e33483.
- [13] Larsen PL, Albert PS, Riddle DL (1995) Genes that regulate both development and longevity in *Caenorhabditis elegans*. *Genetics* 139: 1567–1583.
- [14] Morley JF, Morimoto RI (2004) Regulation of longevity in *Caenorhabditis elegans* by heat shock factor and molecular chaperones. *Mol Biol Cell* 15: 657–664.
- [15] Maduro MF, Gordon M, Jacobs R, Pilgrim DB (2000) The UNC-119 family of neural proteins is functionally conserved between humans, *Drosophila* and *C. elegans*. *Journal of Neurogenetics* 13: 191–212.
- [16] Gems D, Riddle DL (2000) Genetic, behavioral and environmental determinants of male longevity in *Caenorhabditis elegans*. *Genetics* 154: 1597–1610.
- [17] Hosono R (1978) Sterilization and growth inhibition of *Caenorhabditis elegans* by 5-fluorodeoxyuridine. *Experimental Gerontology* 13: 369–374.
- [18] Reinsch CH (1967) Smoothing by Spline Functions. *Numerische Mathematik* 10: 177.
- [19] Herndon LA, Schmeissner PJ, Dudaronek JM, Brown PA, Listner KM, et al. (2002) Stochastic and genetic factors influence tissue-specific decline in ageing *C. elegans*. *Nature* 419: 808–814.
- [20] Vaupel JW, Carey JR, Christensen K, Johnson TE, Yashin AI, et al. (1998) Biodemographic trajectories of longevity. *Science* 280: 855–860.
- [21] Johnson TE, Wu D, Tedesco P, Dames S, Vaupel JW (2001) Age-specific demographic profiles of longevity mutants in *Caenorhabditis elegans* show segmental effects. *J Gerontol A (Biol Sci Med Sci)* 56: B331–339.
- [22] Baeriswyl S, Diard M, Mosser T, Leroy M, Manière X, et al. (2009) Modulation of aging profiles in isogenic populations of *Caenorhabditis elegans* by bacteria causing different extrinsic mortality rates. *Biogerontology* 11: 53–65.

- [23] Curtsinger JW, Fukui HH, Townsend DR, Vaupel JW (1992) Demography of genotypes: failure of the limited life-span paradigm in *Drosophila melanogaster*. *Science* 258: 461–463.
- [24] Carey JR, Liedo P, Vaupel JW (1995) Mortality dynamics of density in the Mediterranean fruit fly. *Exp Gerontol* 30: 605–629.
- [25] Weitz J, Fraser H (2001) Explaining mortality rate plateaus. *Proc Natl Acad Sci USA* 98: 15383–15386.
- [26] Steinsaltz D, Evans SN (2004) Markov mortality models: implications of quasistationarity and varying initial distributions. *Theoretical Population Biology* 65: 319–337.
- [27] Steinsaltz D, Evans SN (2007) Quasistationary distributions for one-dimensional diffusions with killing. *Transactions of the American Mathematical Society* 359: 1285–1324.
- [28] Cox C (2008) The generalized F distribution: an umbrella for parametric survival analysis. *Statist Med* 27: 4301–4312.