nature | methods

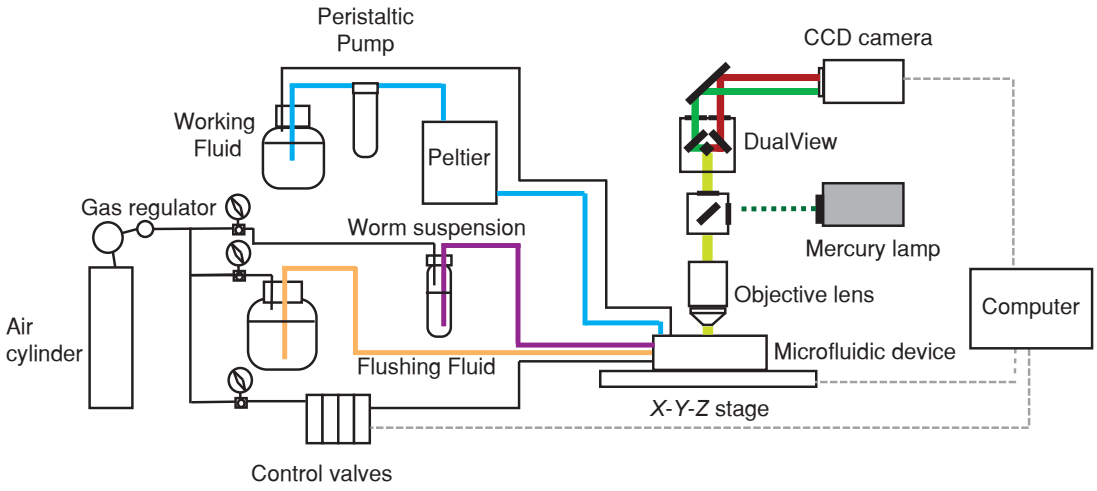
Autonomous screening implicates new genes in synaptogenesis of *C. elegans*

Matthew M. Crane, Jeffrey N. Stirman, Chan-Yen Ou, Peri T. Kurshan, James M. Rehg, Kang Shen, and Hang Lu

Supplementary Item	Title or Caption
Supplementary Figure 1	Schematic showing the entire system required for device operation
Supplementary Figure 2	Microfluidic device design, fabrication, and operation
Supplementary Figure 3	System operation flow diagram
Supplementary Figure 4	The creation of a ground-truth library
Supplementary Figure 5	The image processing stages for synapse identification
Supplementary Figure 6	Training the synapse classifier first stage
Supplementary Figure 7	Training the synapse classifier second stage
Supplementary Figure 8	Receiver operating characteristic curves for each of the stages of the synapse classifier
Supplementary Figure 9	Receiver operating characteristic curve for the wildtype vs *lin-44*$^{-/-}$ classifier
Supplementary Table 1	Automated screening steps and time required
Supplementary Table 2	List of mutants identified during screening
Supplementary Note 1	System operation
Supplementary Note 2	Computer vision and phenotyping
Supplementary Note 3	Machine learning overview
Supplementary Note 4	Screening approach
Supplementary Software	Software

Nature Methods: doi:10.1038/nmeth.2141

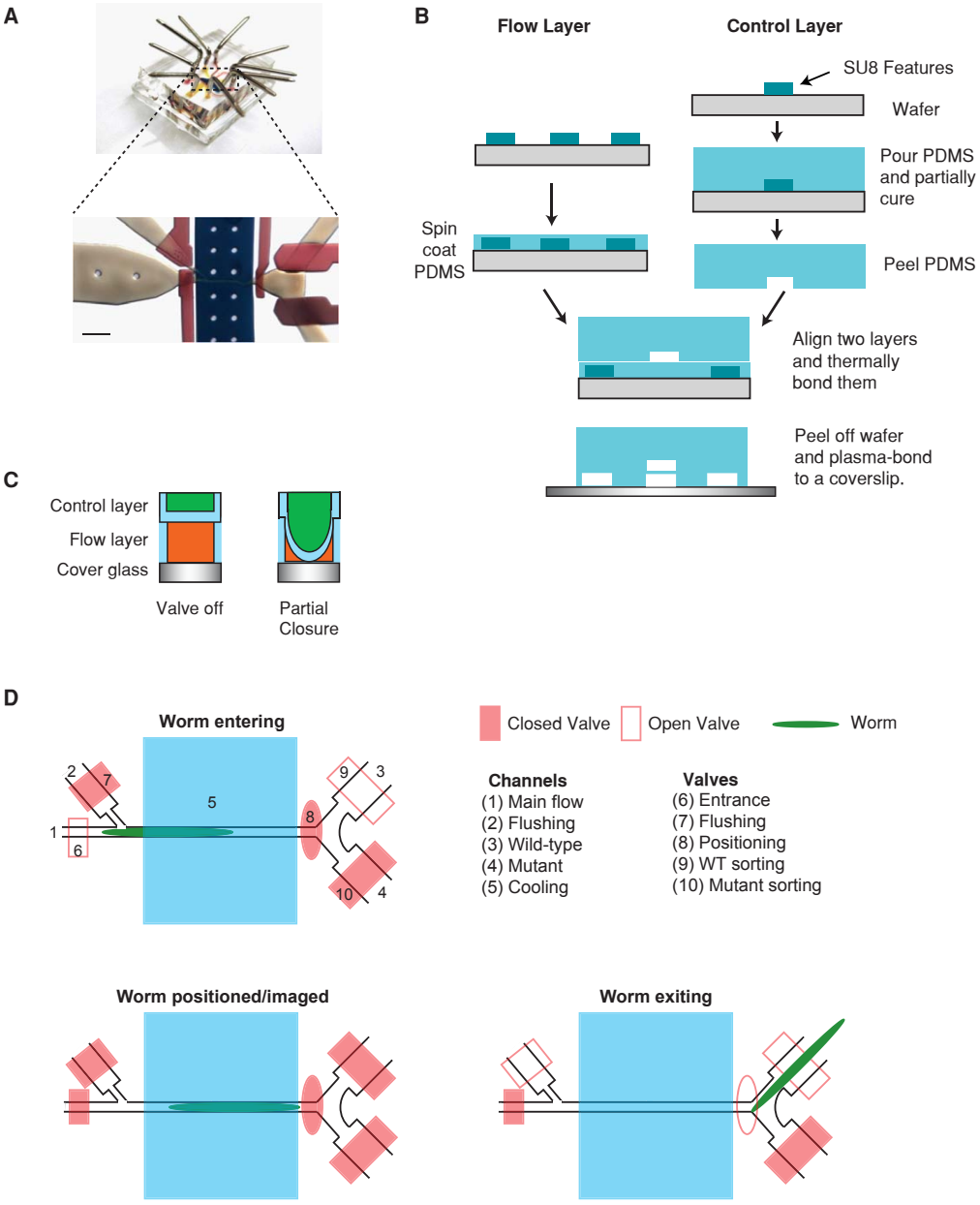
Supplementary Figure 1



Schematic showing the entire system required for device operation. The microfluidic device is placed under a conventional compound epifluorescent microscope. To enable closed loop, automated operation, external components were designed to handle injection of the worm solution, valve control, flushing and animal cooling.

Nature Methods: doi:10.1038/nmeth.2141

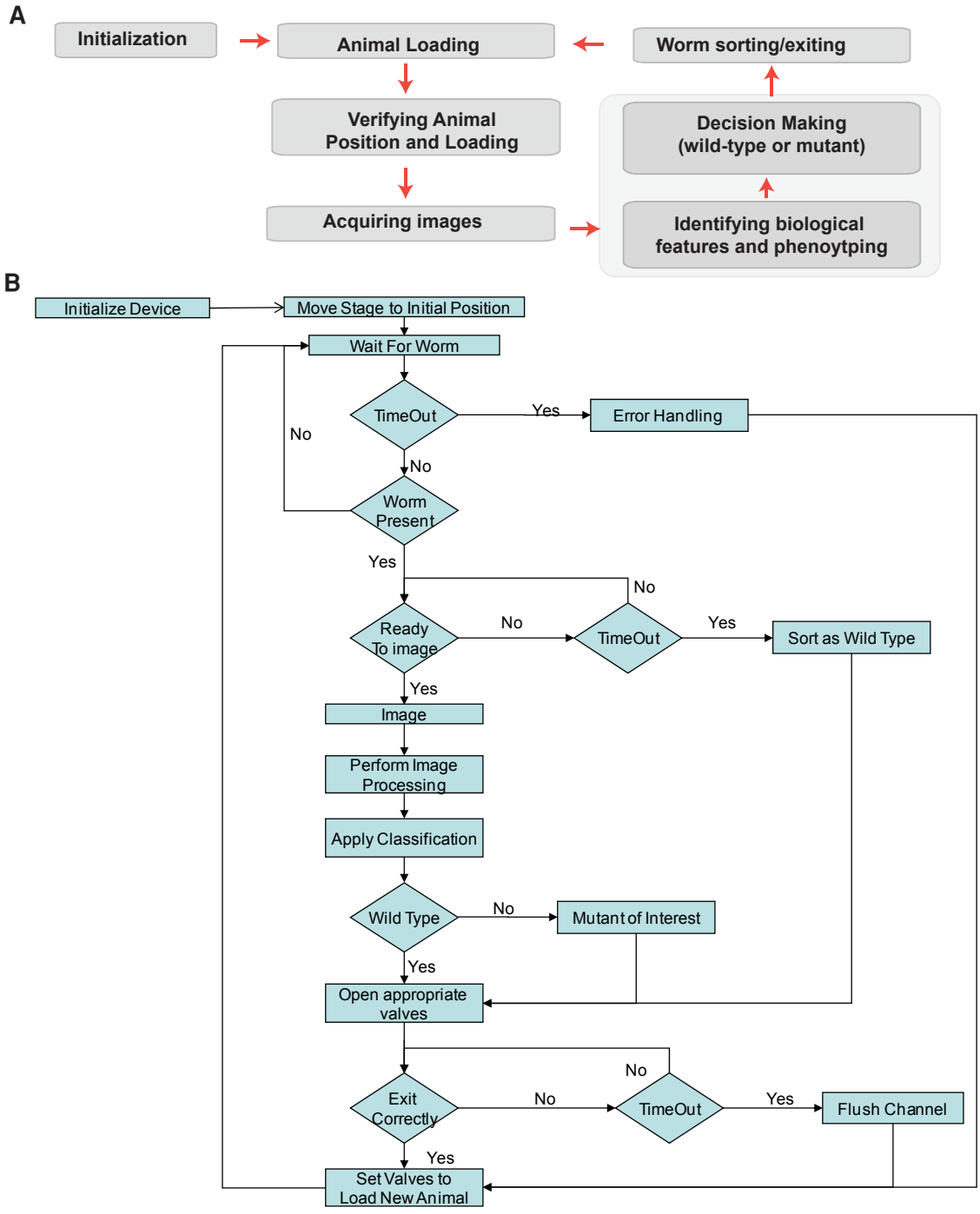
Supplementary Figure 2



Microfluidic device design, fabrication, and operation. (A) A photograph of the dye filled device. Pins have been inserted into the appropriate fluid entrances. Inset is a magnified view of the device showing the critical operating region. Valves are filled with red dye, the cooling channel is filled with blue, and the yellow is the fluid flow channel for the worms. (B) The device fabrication procedure using conventional two-layer soft-lithography. (C) A cross-section showing the valves used in the device. The flow layer has a rectangular cross-section that allows a small amount of flow even when closed. This flow guides the animals into the imaging zone, but the small opening is too small for the animals to pass by. (D) Cartoon showing the sequential stages of animal loading, imaging and sorting.

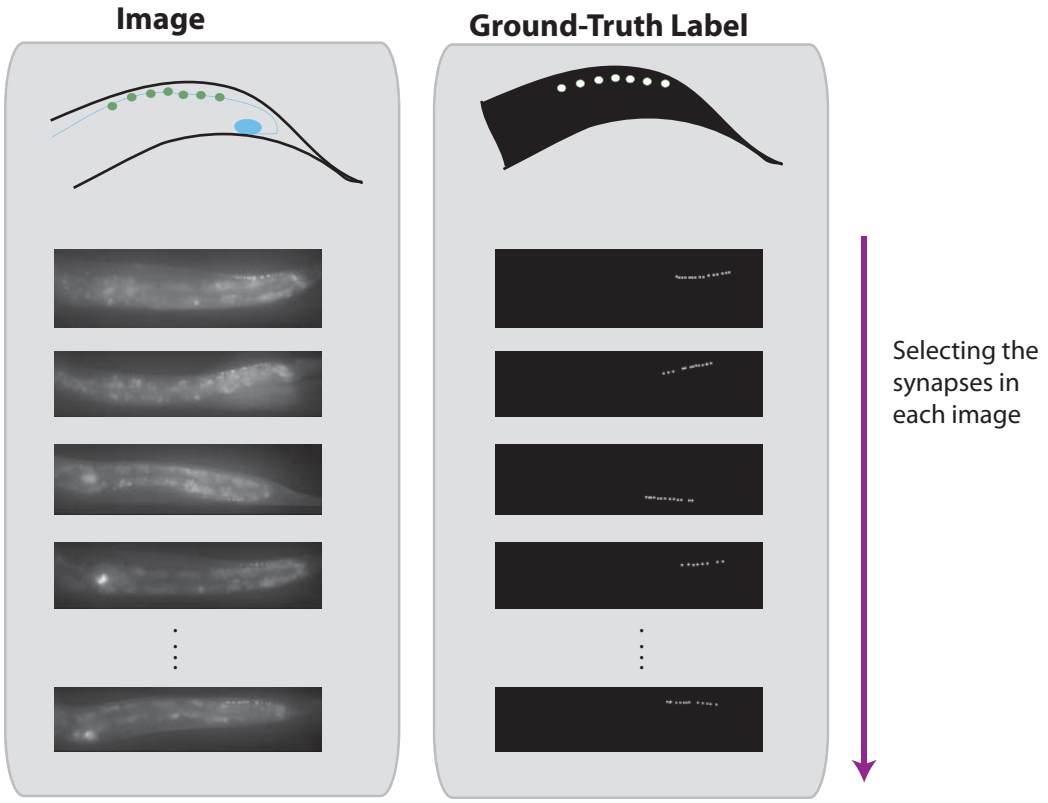
Nature Methods: doi:10.1038/nmeth.2141

Supplementary Figure 3



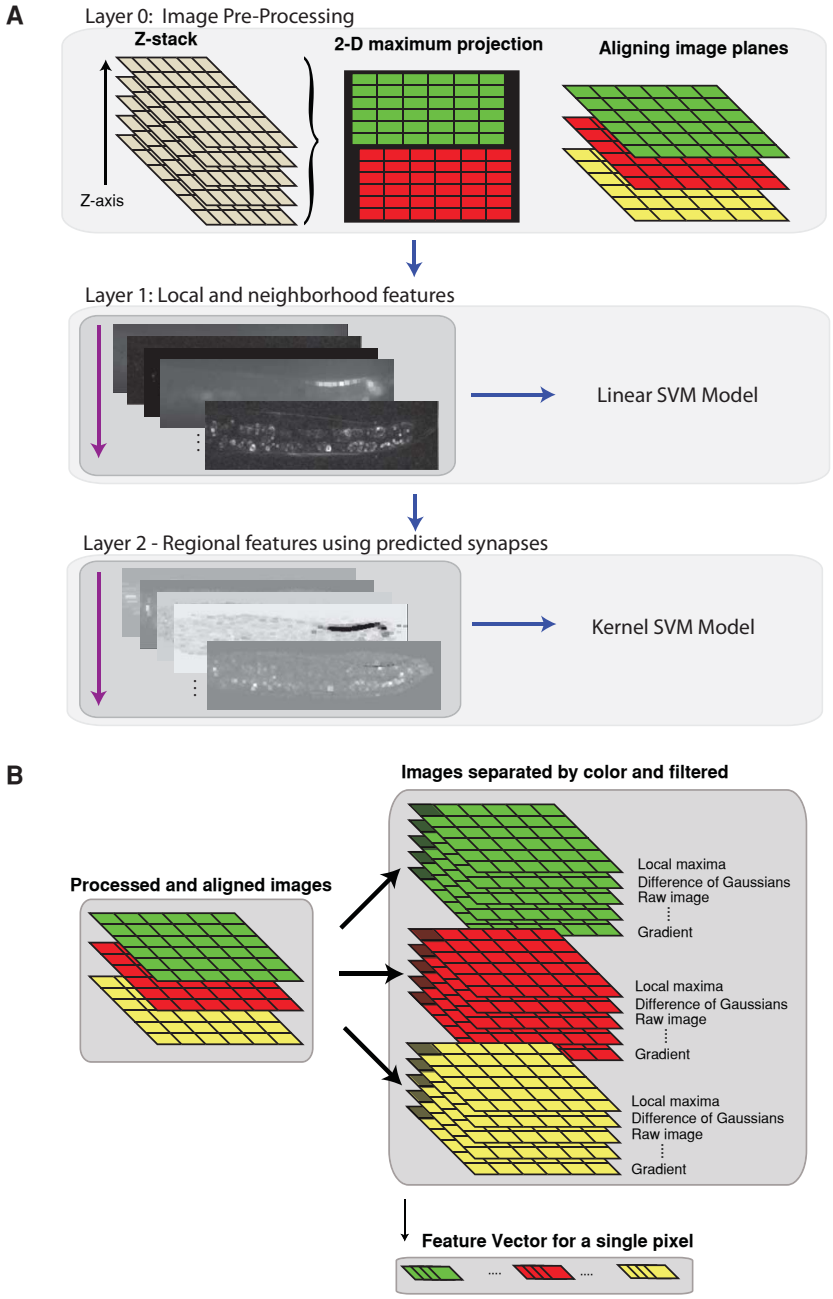
System operation flow diagram. (A) System operation overview. Following initiation of the system, it enters a closed loop wherein animals are loaded, imaged and screened without interruption. The computer vision and machine learning portions (grouped in the lower right) allow self-directed mutant discovery. (B) Detailed flow diagram with the appropriate error handling. Error handling enabled continuous operation and minimized the errors requiring human intervention.

Supplementary Figure 4



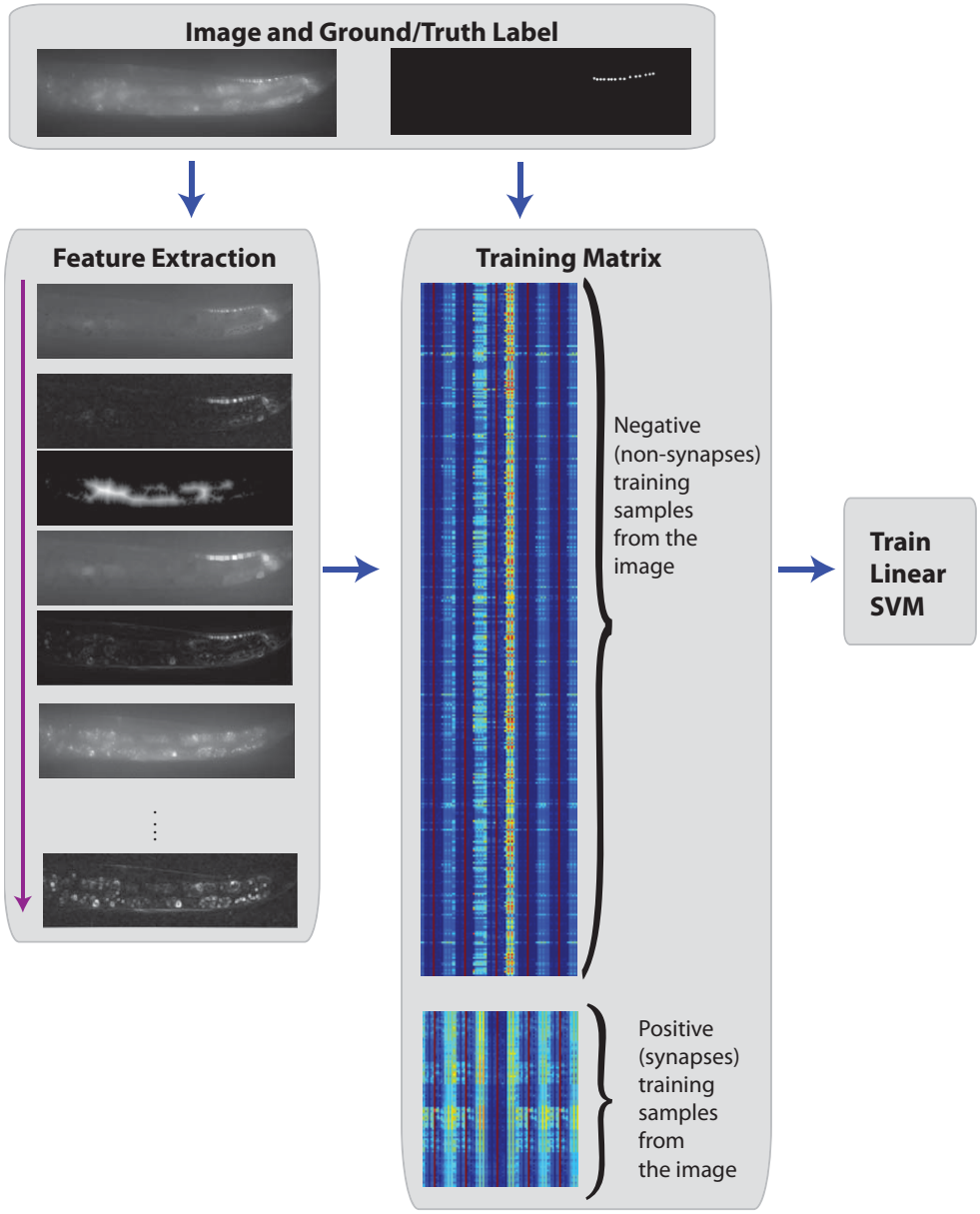
The creation of a ground-truth library. Z-stacks of numerous strains were acquired in the microfluidic device and saved for future use. These synapses within the images were labeled to create ground-truth libraries.

Supplementary Figure 5



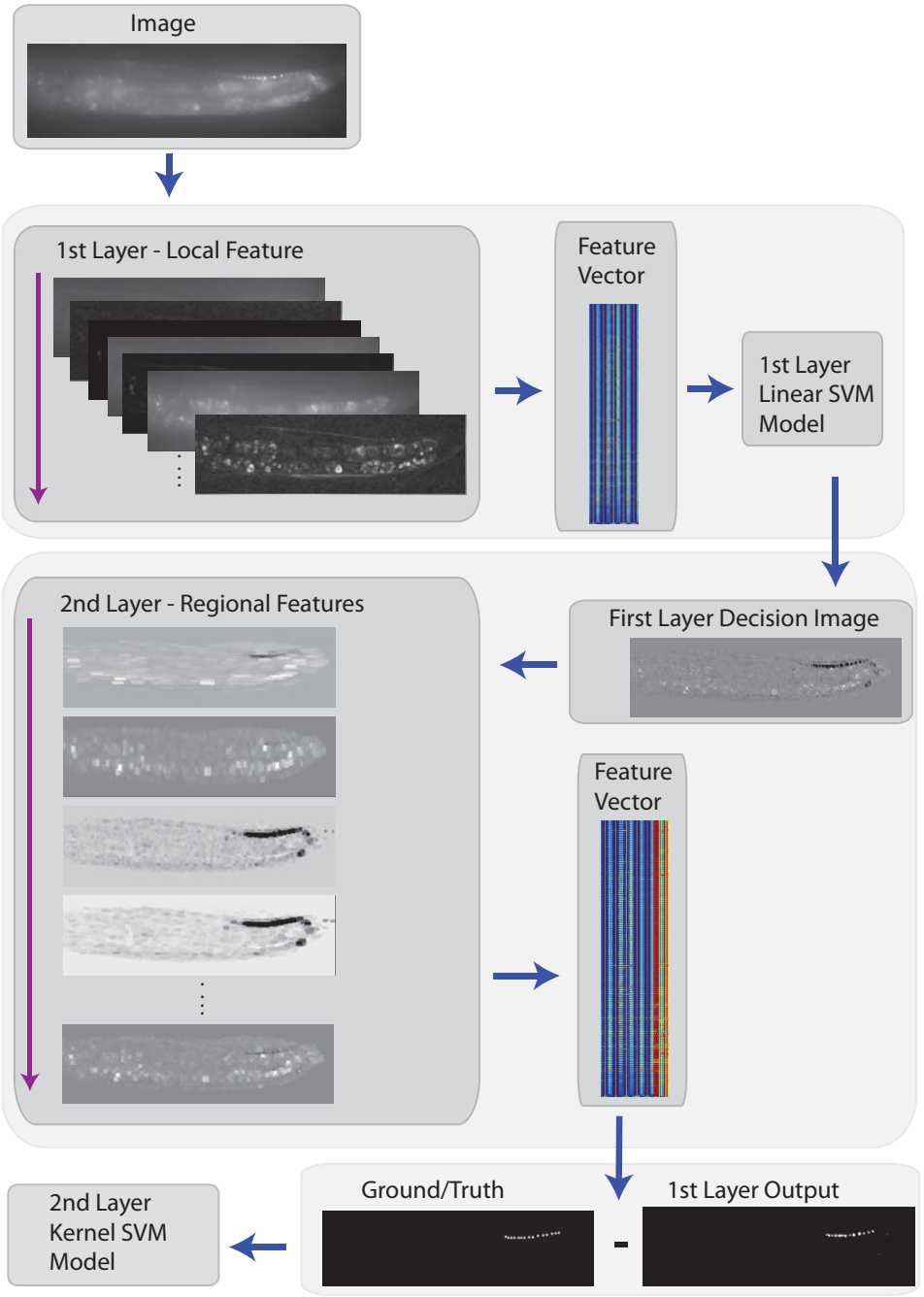
The image processing stages for synapse identification. (A) First, the acquired z-stack is processed to reduce the dimensionality and increase the SNR. Then, local features are used to identify probable synapse locations in a rapid manner. Thirdly, the probable synapses and the original post-processing image are used to extract regional features and identify the actual synapse locations. (B) A schematic showing the process of extracting the local features for the rapid, first pass classification. The initial image containing the green, red and ratio image on the left is separated and used to calculate a sequence of features for each of the images. Finally, each pixel has a vector of features that can be used to predict the likelihood that it is part of a synapse.

Supplementary Figure 6



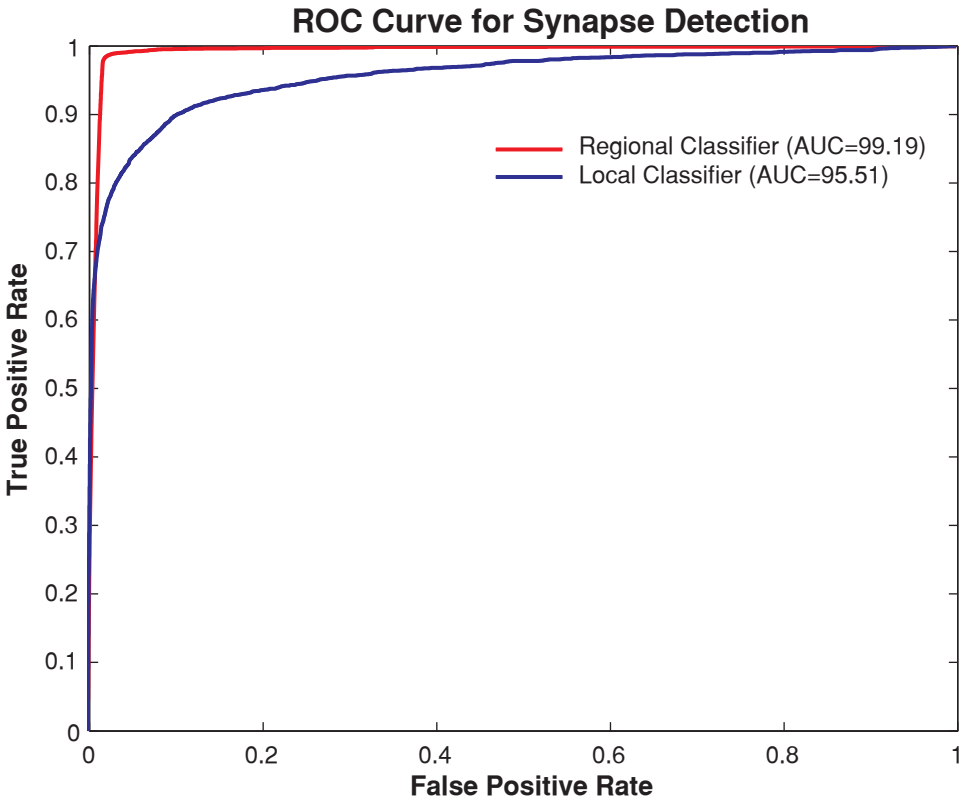
Training the synapse classifier - first stage. Local features are extracted from each of the ground-truth labeled images. Within the images, positive (synapses) and negative (non-synapses) points are selected based on the ground-truth labels. These points, and the corresponding features are used to train a linear-SVM classifier to distinguish between synapses and non-synapses.

Supplementary Figure 7



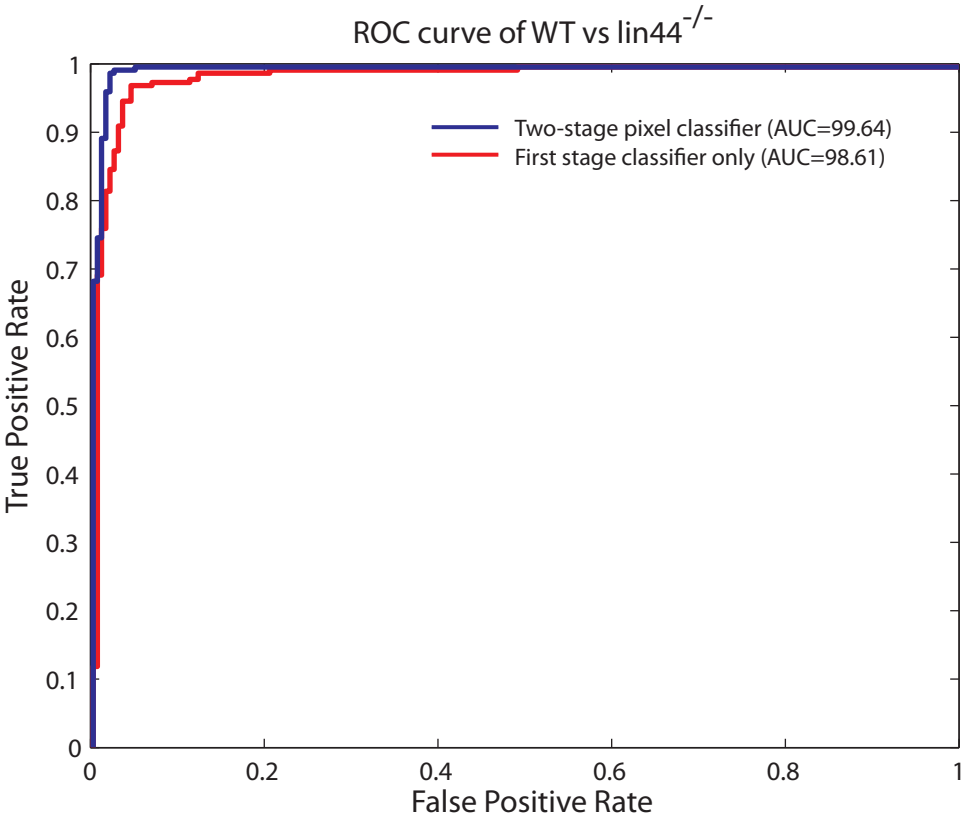
Training the synapse classifier - second stage. The first layer (Supplementary Figure 6) is applied to each of the ground-truth labeled images, and used to obtain an estimate of whether an individual point belongs to a synapse. This estimate is used to extract features based on the spatial relationship between potential synapses. This feature vector is then combined with the feature vector from the first layer and trained using an RBF kernel-SVM.

Supplementary Figure 8



Receiver operating characteristic curves for the two synapse classifiers calculated using the pixel-by-pixel method on the ground-truth labeled library. The error rates were determined using 5-fold cross-validation and a grid-search for parameter optimization for each classifier. Because it is a rare-search problem, the 3,000 points in each image most likely to be synapses were used to determine false and true-positive rates. This accentuated the performance differences between the two classifiers.

Supplementary Figure 9



Receiver operating characteristic curve for the wildtype vs *lin-4*$^{-/-}$classifier. Two curves are plotted showing the performance of the whole animal classification when either the both stages of the pixel classifier are applied to identify synapses, or just the first stage employing local information. The individual pixel by pixel performance of these classifiers is shown in Supplementary Figure 8. A brute force grid search was applied to determine the optimal weights for the first stage of the pixel classifier that gave the best whole animal performance. This was then used to determine the optimal weight of the second stage pixel classifier. This was done using 5-fold cross-validation and a further grid-search for the cost and gamma at each stage. As can be seen, the performance of the pixel level synapse classifier impacts the whole animal classification, and the two-stage classifier performs noticeably better than the one-stage. The error rates were determined using 5-fold cross-validation and a grid-search for parameter optimization.

Supplementary Table 1

Step	**Time**
Wait for worm to load	1s
Pause to insure immobilization	1s
Check Head/Tail orientation	.5s
Sorting	.5s
Wait for worm to load	1s
Pause to insure immobilization	1s
Check Head/Tail orientation	.5s
Pause to insure immobilization	1s
Acquiring 60 um stack with 2 um steps	4s
Image processing to identify synapses	2s
Phenotyping and decision making	.5s
Sorting	1s
Subtotal	**14**

A table containing the sequential imaging and screening steps, and the time required for each stage. Actual time per stage fluctuated slightly depending on animal loading/sorting times.

Supplementary Table 2

Strain Name (allele name)	Screen Method	Potential pathway
GT81 (a081)	Discriminative	Patterning
GT82 (a082)	Discriminative	Patterning
GT83 (a083)	Discriminative	Patterning
GT84 (a084)	Discriminative	Trafficking
GT85 (a085)	Discriminative	Novel
GT86 (a086)	Discriminative	Patterning
GT87 (a088)	Discriminative	Patterning
GT88 (a088)	Discriminative	Patterning
GT89 (a089)	Discriminative	Patterning
GT90 (a090)	Discriminative	Trafficking
GT98 (a098)	Outlier	Assembly
GT100 (a100)	Outlier	Patterning
GT102 (a102)	Outlier	Trafficking
GT103 (a103)	Outlier	Patterning
GT104 (a104)	Outlier	Patterning
GT105 (a105)	Outlier	Expansion
GT106 (a106)	Outlier	Patterning
GT107 (a107)	Outlier	Trafficking
GT108 (a108)	Outlier	Patterning
GT109 (a109)	Outlier	Trafficking
GT112 (a112)	Outlier	Patterning
GT114 (a114)	Outlier	Patterning
GT115 (a115)	Outlier	Patterning
GT117 (a117)	Outlier	Trafficking

A list of the mutants that were characterized with greater detail, along with the predicted pathway and the screening approach used to identify them.

Supplementary Note 1

System Operation

In addition to the development of a microfluidic device for screening, considerable engineering and design were required to create the external systemic components to allow for automated sorting. This required a closed loop control system and the development of specific hardware to interface with and control the on-chip components. Creating external components that would allow computerized control of on-chip components was necessary. The system configuration including the external components can be seen in Supplementary Figure 1.

Earlier efforts demonstrated that automated sorting was technically feasible[2]; these efforts, however, lacked error handling and thus required nearly constant supervision during operation. Because we needed to run the system for a very long period during the screen, a comprehensive framework was developed for handling errors robustly. This includes an external macroscale control components to be integrated with the microfluidic device to allow closed-loop control, an error handling routines to reduce the need for operator intervention for the closed loop control, and optimization of the operational sequence to minimize the amount of time spent per animal. The general framework for system operation is shown in Supplementary Figure 3A. The entire system was coded within Matlab. External calls were to the machine learning libraries and a dll for valve control operation. Due to potential stability issues with Matlab operation, however, we would encourage other researchers to focus on the open microscopy environment and code in a more robust language such as Java.

One of the critical components of creating a robust system that can screen thousands of animals for hours at a time was the development of error handling routines. Through careful design and optimization of the device and screening protocols, it is possible to reduce the number of errors during normal operation, but it is not possible to eliminate them. A system for extended screening, therefore, relies on extensive error handling to deal with cases for when system performance is non-optimal.

1) Flow errors. Grouped into these is when worms take too long to arrive or exit.
2) Orientation and image quality errors.
3) Computational errors.

Supplementary Figure 3B shows the flow diagram with the error handling and decision making steps.

Microfluidic Device Operation

During the loading process (Supplementary Figure 2D), the openings near the wall underneath positioning control valve allow moderate fluid flow through the device. This serves to push the worm through the device and into the imaging channel. Once a worm is pulled into the imaging channel, the flow pushes it up against the position control valve, which stops it. Because the gaps under the valve are substantially smaller than the worm, it is unable to move further forward, and is stopped. At this point the presence of the worm within the imaging channel causes a significant change in the fluidic resistance, and creates a corresponding decrease in the flow rate. The entrance valve behind the

animal is closed to prevent additional animals from entering, and the animal is imaged. Following imaging and decision-making, the appropriate sorting valve is opened in addition to the positioning valve. The flushing valve (in addition to the off-chip pinch valve controlling the flushing fluid) is opened to push the imaged worm out of the imaging region in <300ms. This process is then repeated for the next worm.

In order to ensure that each animal was released into the appropriate exit channel in a consistent amount of time regardless of the animal size, or the fluidic resistance of the other regions of device, a flushing system was created to remove animals. This uses a high pressure (typically twice the pressure of the injection pressure) to push the animals out of the imaging channel. Using both on-chip and off-chip valves, this high-pressure flow is kept isolated from the system until an animal needs to be released. At this point the loading regulator valve is closed and the high pressure valve opened. This forces the majority of the flow through the imaging channel, and rapidly pushes the imaged worm into the appropriate channel. The flushing channel greatly improves the robustness and consistency of the device performance for the following reasons:

1) Animals exit in the same amount of time despite of the non-uniformity in animal size (due to mutagenesis).
2) Flushing flow eliminates changes in the flow rate through the entire device caused by resistance changes elsewhere in the device, e.g. a worm of a slightly large size or a piece of debris. This is important because the device relies on pressure driven flow, and a change in resistance anywhere in the device affects the flow rate through the entire system.

The microfluidic system requires a method of immobilization that would be capable of completely immobilizing the animal for several seconds in a reliable manner. Rather than using conventional method of anesthetics, which can potentially affect synapse expression and morphology, we chose to briefly cool the animals. Cooling the animals to a low temperature (<5°C) was found to immobilize the animals in a manner that was completely reversible, and as long as it is short-term, appears to have no effects on synapse morphology[2], egg-laying ability, lifespan or behavior.

Screening Process Flow and Speed

One of the critical elements of the system design is the throughput. Doing genome-wide forward genetic screens quickly becomes a relatively daunting numbers game, and one where throughput quickly becomes a significant. Previous work[2] has demonstrated that the throughput was significantly affected by type of sorting and resolution required. For a high magnification sorting looking for features that were only present in either the head or tail, throughput was limited to ~60 animals per hour. To reduce the hurdle for performing genome scale, saturated screens, bottlenecks were identified and eliminated. The final process flow, along with the time required for each stage is presented in Supplementary Table 1. In this scenario, following worm loading the animals would be classified as the correct orientation, or the incorrect orientation. These optimizations in the screening protocol and the device allowed for a nearly four-fold increase (>220 animals per hour) in throughput relative to the first generation system.

Supplementary Note 2

Computer Vision and Phenotyping

We developed a computer vision framework to automatically handle images of animals in order to make decisions based on the fluorescent patterns present. This represents a significant advance over previous methods of phenotyping *C. elegans*[2,24-26]. The challenges presented by our problem and many screens today are largely a result of the challenging biological nature of the problem - the identification of small (subcellular) features that are extremely dim and are present in an extremely noisy environment. Compared to our problem, phenotyping and sorting of fluorescent patterns can be a simple task if the fluorescent signals are large and extremely bright relative to the surrounding tissues as is often the case with cellular-scale reporters[2,27,28]. This is because the signal to noise ratio is high, which makes it simple to identify the objects of interest by using a simple thresholding[2,27]. By comparison, the framework presented here needs to be able to deal with the small size, limited fluorescence and high background noise; we therefore developed a multi-stage computer vision framework:

1) A computer vision system to identify the primary objects of interest (synapses). This is a multistage process that separates the synapses from the rest of the image that often includes objects of similar shape and fluorescence intensity.
2) A second stage to identify landmarks in the image and then using the extracted synapses and the landmarks to quantify specific phenotypical features. These features are used to project each animal (and class of animals) onto the phenotypical space for classification as wild-type or mutants.

In order to create a trained computer vision system for identifying synapses and screening animals, a ground truth library was created (Supplementary Figure 4). In conventional computer vision problems there are a significant number of curated databases that are used for both training purposes[29-31], as well as objective measures of performance. To create these databases, we labeled and annotated images manually.

Images were labeled both holistically, and to identify specific features within each of the images. Each of the images was labeled as head, tail, or mid-body. This labeling was used to help optimize the device operation by rejecting animals that were of the incorrect orientation. In all of the animals that were labeled as tail animals, the synapses were then identified within the image (Supplementary Figure 4).

Synapse Identification

The synapse identification is the most challenging of the steps due to the low signal-to-noise ratio (SNR). Furthermore, the synapse identification is complicated by multiple technical requirements in addition to the low SNR. Challenges and requirements for the system performance can be separated into three distinct areas:

1) Throughput: Image processing and image recognition must occur in near real-time. Decision making must occur on-line immediately following the image acquisition, and each additional second reduces the screening throughput. The objective was to create an image processing system that required less time than the image acquisition, which is limited by the state-of-the-art hardware system.

2) Image properties
    a. The images had a low SNR due to the high levels of autofluorescence, small synapses and fat granules with similar sizes and fluorescent intensities. The synapses also have limited numbers of fluorophores and photo-bleach quickly. Therefore detection needs to be sensitive and the phenotyping needs to be discriminative and accurate.
    b. Animals enter with variations in rotational orientation. Because a cuticle surrounds each animal, the body of the animal can act as a lens and imaging through the organism can create distortions in the expression pattern and image. Therefore the code needs to be able to handle variations in synaptic patterns due to animal orientation.
    c. The small depth of field means that the information of interest (synapses) are often at different focal planes, and thus a pseudo 3-dimensional image must be acquired.
3) Nonuniformity in animal population: Dealing with a mutagenized population of animals introduces a significant degree of variation into the animal appearance. This includes significant variation in size (length and width) as well as changes in levels of autofluorescence or size and distribution of fat granules. Even more significant changes to animal appearance, such as increased numbers of embryos, the appearance of blisters on the cuticle, or even animals that have hatched inside the mother are commonly seen.

These factors imposed constraints on the image processing that had to be considered for each of the steps. Specifically, the variations in the images and image quality make it challenging to correctly separate the correct animals from irrelevant features. To deal with these variations, the framework developed performs upfront image processing to reduce the image from a three-dimensional image and to reduce the complexity while maximizing the information present in the image. From these images with reduced complexity, local features are extracted based on the nearby pixels. Larger region-based features are then extracted using both the local features and heuristics applied to the image based on knowledge about the animal position and appearance.

The image processing was separated into three stages (Supplementary Figure 5A):

1) Upfront image pre-processing. This served to reduce the dimensionality of the images while preserving the majority of relevant information. Additionally, the image was separated based on the spectral filter and aligned.
2) Rapid local feature filtering. This was designed to extract features from small neighborhoods surrounding each pixel, and then to use a fast linear-SVM to identify potential synapse locations.
3) Regional feature extraction. Using both features extracted from larger regions and the potential synapse locations and relative positions, a second layer of features was created. To prevent this from becoming a bottleneck, only the locations with a high probability of being actual synapses from the previous step were analyzed using an RBF-kernel SVM.

Image Pre-Processing

During the image acquisition stage, images are acquired at multiple focal planes at regular steps along the z-direction of the animal. For most of these experiments, images were acquired at 1.5 micron steps over 60 microns. This ensured that regardless of the original position of the objective, and the orientation of the animal, the objects of interest (synapses) would be in focus. Although the synapses tend to be relatively in plane, they

are often present in multiple images. This occurs especially frequently when synapses are on opposing sides of the animal. In these cases, even a small rotational change in the animal can result in synapses being in-focus ten microns apart from one-another. This meant that the image processing could not be confined to a single image plane.

The primary challenge was in dealing with a massive amount of information and identifying the synapses within a few seconds. Images were acquired at a 640X480 resolution, and with 40 image planes, this resulted in 12 million pixels. Processing these in a volumetric manner was prohibitively computationally intensive. Therefore, this volumetric image was compressed into a 2-dimensional image by taking the maximum projection at each x-y point along the z-direction. The conversion to a 2-D image using maximum projection reduced the number of pixels required for high level feature extraction, to a much more manageable 300 thousand. Several alternative projection methods such as mean projection, or standard-deviation projection were tested, but none of these provided improved performance.

Following projection, the flattened image was separated into three distinct images, each of which will be used for feature extraction. The use of the DualView™ makes it possible to obtain simultaneous two-color images of the same focal plane (Supplementary Figure 2). Thus, the top half of the flattened image shows the fluorescence in the mCherry channel, while the lower portion shows fluorescence in the GFP channel. Additionally, the imaging process of the DualView™ creates an offset between the different spectral image channels that can vary significantly between individual imaging runs. To correct for this, the two separate images were aligned. Before each screen or imaging run, images were acquired the bright-field mode and separated into a top and bottom image. Using the normalized cross-correlation, the offset between the two images was determined and used for that screening run. Because the absolute intensities can vary, for example increasing across both the red and green channels, a third image was created using the ratio between the green and red channels (Supplementary Figure 5A).This tri-plane image was then used for feature extraction and identification. The use of multiple filters significantly increases the SNR and allows for significantly higher performance.

Rapid local feature filtering

The local features were extracted based on small bounding regions (Supplementary Figure 5B). These features were designed to be calculated in an extremely rapid manner and to accentuate the differences between the small synaptic fluorescent expression pattern and the surrounding tissues. Synapses are small regions of bright fluorescence, surrounded by regions of relatively dimmer fluorescence. Furthermore, the ratio of the fluorescence in the green relative to the red images is higher than in the autofluorescence. The majority of these features were calculated in neighborhoods varying from [3 3] up to [8 8].

Because the initial transformation created a matrix containing three separate images (green, red, and green-red ratio), the set of local features was calculated on each of the individual images (Supplementary Figure 5B). Using the same set of features increases the flexibility of the system. Additional upfront processing to create additional layers could be easily added, and then the same filters applied.

The filters applied included some adapted from the SIFT features[32,33]. These include the local maxima and minima, gradient, local average, median filtering, difference of Gaussians with different Gaussian filters and the local standard deviation of the neighborhood. Applying these filters over different size scales allowed the filters to capture the variability between the fat granules and synaptic expression. Furthermore, calculation of these features is rapid, with all features calculated in <700 msec on a 640X480 image using Pentium Core i7 processor with 4GB of RAM. Following computation of the feature space, the resulting feature matrix was (320X480) or 153,600x115 and applying even a rapid linear SVM to classify all these points would be time consuming (requiring >1s). The computational burden was reduced by only classifying the pixels where either the GFP or mCherry intensity was greater than the mean intensity. This essentially served to only classify points within the animal.

To train the first layer of the synapse classifier, these features were applied to the ground-truth library (Supplementary Figure 6). The library used for synapse training was composed of 129 images where the synapses present in each were labeled. In each of the labeled image, the full set of features was extracted and then the image was separated into positive and negative training samples. Due to the size of the synapses relative to the image, the vast majority of pixels (>99.9%) would be negative training samples. To find the points that would be the most challenging to distinguish from actual synapses, Canny edge detection was performed on GFP image and the points in it were broadened. Then, out of these pixels a random selection of 2,000 pixels per image were selected for negative training. This method allowed a variety of training points to be selected from images that were acquired on different days and in different strains to correct for any experimental variability. All of the positively labeled pixels within each image were added to the training dictionary. Using this dictionary, a linear-SVM was trained using liblinear[23]. A brute force grid search optimization method using 5-fold cross-validation was employed to determine the optimal training parameters. The synapses were weighted three times more than the non-synapses during training. The optimal cost during the grid search was found to be C=16. Because of the large number of features extracted from the image, a linear SVM was found to perform well. For comparison, an RBF kernel SVM was trained on the same dictionary, and with 5-fold cross validation was found to have slightly better (but nearly identical) performance. The kernel SVM (libSVM), however, was prohibitively computationally intensive, as it required more than an order of magnitude more time for classification.

Regional feature extraction

The second layer of the synapse recognition algorithm serves two purposes.

1) The first purpose is to use the locations of the probable synapses to create extract additional features based on the relative position of the potential synapses to one another.
2) The second is to train a more powerful classifier using just the potential synapse locations as training examples. This classifier has access to the first, local features, in addition to the new features based on the relative locations and image as a whole.

The second layer features use *a priori* knowledge about the biological structure of the synaptic expression and probable mutant expression patterns to extract 30 additional features. The most strenuous biological constraint is that any and all synapses must be

formed along the axon or dendrite protruding from the cell body. Of course, there could be significant changes in the guidance cues and thus placement of the axon or dendrite, but this is highly unlikely. Furthermore the development of the specific synaptic pattern is constrained by numerous extra- and intra-cellular cues leading to stereotyped expression along the axon in the wild-type animals. Additionally, synapses are more likely to cluster near one another than to be completely isolated.

Conditional random fields (CRF) among other probabilistic models have been employed to optimize image segmentation by using the relative positions of identified features to more accurately segment images[34-38]. A CRF based method was initially implemented but found to be far too slow. The feature extraction for this portion of the method can take at most 1 second, and the CRF based method required tens of seconds. To optimize for speed, while utilizing the *a priori* knowledge about the probable synapse locations, a less computationally method was derived. The decision values for the first layer (local features and linear SVM) were used to create an image that expressed the likelihood that any specific pixel is a part of a synapse. Using this decision image as a template for further image processing, many features similar to the first stage were extracted, but within larger neighborhoods. For example, the local maxima and minima of the decision image was extracted using rectangular neighborhoods varying from [5 5] in steps of 5 up to [15 15]. These larger neighborhoods allowed the information on the density of synapses and whether a particular synapse was isolated.

Rather than employing a probabilistic graphical model to learn the relationship between synapse positions, the knowledge that axons tend to be oriented in relatively straight lines was used to incorporate the surrounding pixel information. A Hough transform was applied to a thresholded version of the decision image to identify the two lines most likely to connect the points identified. The two largest peaks of the Hough transform, corresponding to the most likely lines in the image were identified extracted.

Using these two lines, several morphological features were applied to the image with the raw decision values. These included erosion and dilation using differently sized structuring elements and performing a two-dimensional convolution using lines of different lengths, in the directions of the two primary lines in the image. The directions perpendicular to those identified using the Hough transform were used in addition to primary. If a sequence of low probability synapses were aligned close to one another, these operations would extract that information and allow the second stage classifier to use that information. Likewise, a single point that is far from any other potential synapses is more likely to be an incorrect classification, and these features will reflect that.

For each of the locations, the feature vector from the second layer features is combined with the feature vector from the first layer (Supplementary Figure 7). This creates a feature vector that contains 145 features for each pixel location. Because the linear SVM from the first layer was trained to err on the side of misidentifying non-synapses as synapses, the second layer only looks at locations identified by the first as being potential synapses. This serves several purposes. First of all, it reduces the number of data points dramatically (by 3 orders of magnitude) and thus decreases the computational time required for classification accordingly. Secondly, this decrease in computational time allows the use of a more robust, but slower classification using an RBF-kernel SVM to identify true synapses. Thirdly, it allows the classifier to be trained only on points that

were misclassified in the first layer and thus provide an increase in classification accuracy (Supplementary Figure 7).

Using the ground-truth library of images, this second stage incorporating the regional features was trained. A brute force grid search optimization method using 5-fold cross-validation was employed to determine the optimal training parameters. The optimal parameters identified during the grid search were $C = 2\ and\ \gamma = .0625$.

In order to determine the performance of the synapse classifiers, pixel performance statistics were calculated for each of the two methods (Supplementary Figure 8). Performance characteristics were determined using each pixel within the image, and by whether the individual pixel should be a synapse or not. The library of ground-truth labeled images was used for training both of the classifiers. Because the two-stage classifier incorporating the regional features is dependent on the first layer (local features), the image library of images was divided into training and testing sets for 5-fold cross validation. Both classifiers were trained and tested on the same image set. During each of the cross-validation performance tests a grid-search was performed to determine optimal training parameters for the training. Because identifying synapses is a rare-search problem, the vast majority of pixels were correctly rejected using both classifiers. Using all of these data points obscures the performance differences between the two methods, so for each image only the 3,000 pixels most likely to be synapses were used to determine classifier performance. Employing this method, the receiver operator characteristic (ROC) curve in Supplementary Figure 8 was calculated. An ROC curve is a generic way to measure classifier performance, and is calculated by comparing the true positive rate to the false positive rate, and is commonly used to compare different classifiers to determine which performs better. As can be clearly seen in the figure, the two-stage classifier that incorporates regional features performs significantly better than the classifier only using the local features.

Phenotypical Feature Extraction

Once the synapses have been located and identified, it becomes important to identify other features, external to the synapses themselves that can be used for extracting quantitative features. In some cases, this could be the distance of synapses to the soma of a specific cell, or the number of synapses that are clustered into specific areas in the animal. Therefore, the landmarks must be identified. In identifying the synapses of DA9, because no fluorescent reporters were present in the area nearby the synapses, several alternative landmarks were identified and used. The landmarks identified were the midline of the animal and the end of the intestine. Because synapses are only supposed to be located on the dorsal side of the animal, correctly determining whether all the synapses are on the same side is important, and is done using the animal midline. This can be done relatively easily because the autofluorescence of the animal is significantly greater than the surrounding areas. The midline is determined by multiplying the red and green channel images, and thresholding for any locations greater than the mean. This image was then thinned to identify the midline. Using both spectral images improved the SNR and reduced the odds of misidentifying the midline.

The second landmark used was the end of the intestine. Whereas the first landmark serves to identify whether individual synapses are located on the ventral or dorsal side of the

body, the second feature helps to whether synapses are in the asynaptic regions posterior to the DA9 cell body. The stereotyped position of the intestine and the relatively strong levels of fluorescence make the intestine an appealing landmark. The end of the intestine corresponds very similarly to the location of the beginning of the asynaptic zone in wild-type animals.

For each of the synapses located in the image, the location relative to the two landmarks was identified. Then, features concerning the synapses and its relationship to the other identified synapse were extracted. These include features such as the distance to the nearest and furthest synapse. Additionally, the number of synapses located within a small radius around the synapses is extracted. Following these, features regarding the individual synapse in isolation are extracted. These included features such as the size and fluorescent intensity.

The precise features extracted from the images are:

1. Number of synapses
2. Mean gfp fluorescent intensity of all synapses
3. Std of the gfp fluorescent intensity of all synapses
4. Mean ratio of gfp:mcherry intensity of all synapses
5. Std of gfp:mcherry intensity of all synapses
6. Mean synapse area
7. Std of synapse area
8. Median synapse area
9. Max synapse area
10. Min synapse area
11. Mean distance between all synapses
12. Median distance between all synapses
13. Mean distance of each synapse to synapse nearest it
14. Std of the distance of each synapse to synapse nearest it
15. Median distance of each synapse to synapse nearest it
16. Max distance between any two synapses
17. Min distance between any two synapses
18. Mean distance between each synapse and the end of the intestine
19. Std of the distance between each synapse and the end of the intestine
20. Median distance between each synapse and the end of the intestine
21. Mean number of synapses on each side of the midline
22. Std of the number of synapses on each side of the midline
23. Percentage of synapses that are ventrally located
24. Mean distance of ventral synapses to the end of the intestine
25. Sum of the distance of ventral synapses to the end of the intestine
26. Percentage of synapses that are dorsally located
27. Number of synapses that are dorsally located
28. Mean distance of dorsal synapses to the end of the intestine
29. Sum of the distance of dorsal synapses to the end of the intestine
30. Percentage of synapse located after the end of the intestine

Supplementary Note 3

Machine Learning Overview

For this work, we exclusively used Support Vector Machines (SVM) for our learning and classification purposes. For other purposes, there are extensive reviews in both journals and textbooks, so we only include a brief overview of the approach.

SVMs are a classification method based around the idea of identifying the hyperplane that best separates two sets of points. The underlying theory was developed in the 1960's, but it was only in the late 90's that they become popular in a machine learning context. The simplest way of understanding SVMs is that they attempt to find a line that best separates to sets of points, and maximizes the margin separating the classes on each side of the line (hyperplane)[19]. The name is derived from the points in each class that are located along the margin and called support vectors.

The general form of SVM is[19]:

$$g(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i y_i \langle x, x_i \rangle$$

where the classification rule is $sgn(g(x))$ and SVM optimization requires the minimization of

$$\sum_{i=1}^{N} \left(1 - y_i g(x_i)\right) + \lambda \|\beta\|^2$$

This SVM formulation is a linear classifier, but can be easily transformed into a non-linear classifier using the kernel trick[20]. This allows the data to be mapped into a higher order space, where separation is often performed better, and thus allows significantly increased classification performance. Because the formulation uses a dot product, it can be replaced by any kernel that is positive and semi-definite[21]. Several common kernels are frequently used, and in this work we used the Gaussian radial basis function (RBF) kernel when not using a linear classifier. This is formalized as:

$$k(x, x_i) = e^{(-\gamma \|x - x_i\|^2)}$$

$$g(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i y_i k(x, x_i)$$

One of the advantages of using support vector machines for pattern recognition applications is the large number of curated libraries. Rather than requiring users to reinvent the wheel, these libraries are cutting edge implementations that are extremely fast. Among the different libraries for SVM use are: SVM-light, libsvm, liblinear, MLPack, flssvm, Shark, dlib, and the Matlab default. Many of these libraries were written for specific languages, but have been ported so that they can be used with most programming languages. For this work we used both the libsvm[22] and liblinear[23] libraries,

compiled and then accessed from Matlab. They were selected based on speed and ease of use.

Supplementary Note 4

Screening Approach

Performing a forward genetic screen requires several steps prior to running the actual screen and extracting novel mutants. We first phenotyped the wild-type population containing the fluorescent reporter. This required imaging a sufficient number of *wyIs85* animals under the same conditions as the eventual screen, and then using those images to extract the phenotypical features of interest.

We then determined the method of screening and classifying animals as either mutants of interest or wild-type. The two choices were a trained classifier using the wild-type and a known mutant for training, or an outlier detection screen that looks for anything that seems different from the wild-type population. The discriminative screen used both wild-type animals and a known mutant to train a classifier to identify phenotypical descriptors that maximize the differences between the populations. To screen more broadly for novel mutant classes, we use an outlier-detection scheme, where the phenotypical distribution of the wild-type population was modeled, and animals with a low likelihood of belonging to the population were sorted as mutants (Fig 2B).

Because the correct measure of the classifier performance how well it is able to correctly classify whole animals based on their phenotypical features, the individual stages of the classifier must be optimized with that in mind. This may include weighting synapses and non-synapses differently at the first and second stages of the pixel classifier. To measure the performance of the whole animal classification, and to evaluate the need of using a two-stage as opposed to a single stage pixel classifier, ROC curves of the discriminative classifier were generated. To do this, the first stage was optimized to maximize the area under curve (AUC), and then process was repeated with the two-stage classifier, using the optimized first-stage classifier. The performance of the discriminative classifier is shown in Supplementary Figure 9. To evaluate how important the two-stage synapse classifier was, two ROC curves were generated using the same set of wild-type and *lin-44*$^{-/-}$ images mentioned in the main text.

1 Doitsidou, M., Flames, N., Lee, A. C., Boyanov, A. & Hobert, O. Automated screening for mutants affecting dopaminergic-neuron specification in C. elegans. *Nat Meth***5**, 869-872, (2008).

2 Chung, K., Crane, M. M. & Lu, H. Automated on-chip rapid microscopy, phenotyping and sorting of C. elegans. *Nat Meth***5**, 637-643, (2008).

3 Crane, M. M., Chung, K. & Lu, H. Computer-enhanced high-throughput genetic screens of C. elegans in a microfluidic system. *Lab on a Chip***9**, 38-40, (2009).

4 Rohde, C. B., Zeng, F., Gonzalez-Rubio, R., Angel, M. & Yanik, M. F. Microfluidic system for on-chip high-throughput whole-animal sorting and screening at subcellular resolution. *PNAS***104**, 13891-13895, (2007).

5 Pardo-Martin, C. *et al.* High-throughput in vivo vertebrate screening. *Nat Meth***7**, 634-636, (2010).

6 Jorgensen, E. M. & Mango, S. E. The art and design of genetic screens: Caenorhabditis elegans. *Nat Rev Genet***3**, 356-369, (2002).

7 Gosai, S. J. *et al.* Automated High-Content Live Animal Drug Screening Using C. elegans Expressing the Aggregation Prone Serpin α1-antitrypsin Z. *PLoS ONE***5**, e15460, (2010).

8 O'Rourke, E. J., Conery, A. L. & Moy, T. I. Whole-animal high-throughput screens: the C. elegans model. *Methods Mol Biol***486**, 57-75, (2009).

9 Pardo-Martin, C. *et al.* High-throughput in vivo vertebrate screening. *Nature methods***7**, 634-636, (2010).

10 Chung, K. *et al.* A microfluidic array for large-scale ordering and orientation of embryos. **8**, (2011).

11 Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. High-throughput ethomics in large groups of Drosophila. *Nature Methods***6**, 451-U477, (2009).

12 Dankert, H., Wang, L. M., Hoopfer, E. D., Anderson, D. J. & Perona, P. Automated monitoring and analysis of social behavior in Drosophila. *Nature Methods***6**, 297-303, (2009).

13 Chaumont, F. D., Coura, R. & Serreau, P. Computerized video analysis of social interactions in mice. *Nature Methods***9**, (2012).

14 Bao, Z. *et al.* Automated cell lineage tracing in Caenorhabditis elegans. *PNAS***103**, 2707 - 2712, (2006).

15 Boyle, T., Bao, Z., Murray, J., Araya, C. & Waterston, R. AceTree: a tool for visual analysis of Caenorhabditis elegans embryogenesis. *Bmc Bioinformatics***7**, 275, (2006).

16 Murray, J. *et al.* Automated analysis of embryonic gene expression with cellular resolution in C. elegans. *Nature Methods***5**, 703 - 709, (2008).

17 Murray, J., Bao, Z., Boyle, T. & Waterston, R. The lineaging of fluorescently-labeled Caenorhabditis elegans embryos with StarryNite and AceTree. *Nature Protocols***1**, 1468 - 1476, (2006).

18 Long, F., Peng, H., Liu, X., Kim, S. K. & Myers, E. A 3D digital atlas of C. elegans and its application to single-cell analyses. *Nature methods***6**, 667-672, (2009).

19 Burges, C. J. C. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery***2**, 121-167, (1998).

20 Aizerman, A., Braverman, E. M. & Rozoner, L. I. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control***25**, 821-837, (1964).

21 Boser, B. E., Guyon, I. M. & Vapnik, V. N. in *Proceedings of the fifth annual workshop on Computational learning theory* 144-152 (ACM, Pittsburgh, Pennsylvania, United States, 1992).

22 Chang, C. C. & Lin, C. J.

23 Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X. & Lin, C.-J. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, (2008).

24 Geng, W., Cosman, P., Palm, M. & Schafer, W. R. Caenorhabditis elegans egg-laying detection and behavior study using image analysis. *Eurasip Journal on Applied Signal Processing***2005**, 2229-2240, (2005).

25 Geng, W., Cosman, P., Berry, C. C., Feng, Z. Y. & Schafer, W. R. Automatic tracking, feature extraction and classification of C-elegans phenotypes. *Ieee Transactions on Biomedical Engineering***51**, 1811-1820, (2004).

26 Geng, W., Cosman, P., Baek, J. H., Berry, C. C. & Schafer, W. R. Quantitative Classification and Natural Clustering of Caenorhabditis elegans Behavioral Phenotypes. *Genetics***165**, 1117-1126, (2003).

27 Chung, K. & Lu, H. Automated high-throughput cell microsurgery on-chip. *Lab on a Chip***9**, 2764-2766, (2009).

28 Wang, M., Zhou, X. B., King, R. W. & Wong, S. T. C. Context based mixture model for cell phase identification in automated fluorescence microscopy. *Bmc Bioinformatics***8**, 12, (2007).

29 Davis, J. W. & Keck, M. A. in *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on.* 364-369.

30 Nayar, S. K., Nene, S. A. & Murase, H. in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on.* 2321-2325 vol.2323.

31 Belhumeur, P. N., Hespanha, J. P. & Kriegman, D. J. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on***19**, 711-720, (1997).

32 Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision***60**, 91-110, (2004).

33 Lowe, D. G. in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2* 1150 (IEEE Computer Society, 1999).

34 Plath, N., Toussaint, M. & Nakajima, S.

35 Lafferty, J. & McCallum, A. in *Proc. ICML* (2001).

36 Wallach, H. M. Conditional random fields: An introduction. (2004).

37 Quattoni, A., Collins, M. & Darrell, T. in *In Advances in Neural Information Processing Systems 17* (2005).

38 He, X., Zemel, R. S. & Ray, D. in *In Proceedings of the 9th European Conference on Computer Vision* (2006).