

IMAGE FEATURE EXTRACTION AND NATURAL CLUSTERING OF WORM BODY SHAPES AND MOTION CHARACTERISTICS

Wei Geng, Pamela Cosman, Joong-Hwan Baek¹, Charles Berry², William R. Schafer³
Department of Electrical and Computer Engineering, University of California at San Diego

¹School of Electronics, Telecomm. & Computer Engineering,
Hankuk Aviation University, Koyan City, South Korea

²Biostatistics Unit, UCSD Cancer Center

³Division of Biology, University of California at San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA

ABSTRACT

Genetic analysis of nervous system function relies on the rigorous description of animal behaviors. However, standard methods for classifying the behavioral patterns of mutant *Caenorhabditis elegans* (a microscopic worm) rely on human observation and are therefore subjective and imprecise. Here we describe the application of machine learning and image feature extraction techniques to quantitatively define and classify the behavioral patterns of *C. elegans* nervous system mutants. We have used an automated tracking and image processing system to obtain measurements of a wide range of morphological and behavioral features from videos of representative mutant types. By performing principal component analysis using a selected subset of features, we represented the behavioral patterns of eight mutant types as data clouds distributed in multidimensional feature space. Cluster analysis using the k-means algorithm made it possible to quantitatively assess the relative similarities between worm types and to identify natural clusters among the data. The patterns of similarity identified in this study closely paralleled the functional similarities of the mutant gene products, suggesting that the quantitative image features are an effective diagnostic of the mutants' underlying molecular defects.

KEY WORDS

C. elegans; Natural clustering; Feature selection; Imaging; Shape features

1. INTRODUCTION

Among the organisms most amenable to the genetic analysis of behavior is the nematode (microscopic worm) *Caenorhabditis elegans*. *C. elegans* has a simple nervous system consisting of 302 neurons of known position, cell lineage, and synaptic connectivity [1-3]. Although precise tests for behavioral abnormalities are critical for neurogenetic studies in *C. elegans*, standard visual tests for complex behaviors such as locomotion are typically imprecise and subjective. For example, mutants

displaying abnormal or uncoordinated (“*Unc*”) movement [4-5] are usually observed under the microscope and classified into descriptive categories such as “kinker”, “coiler”, “shrinkers”, “loopy”, “slow”, and “sluggish”. Although mutants with common molecular defects generally have qualitatively similar behavior, the subjectivity inherent in classifying behavioral patterns by eye makes it difficult if not impossible to assess which mutants have genuinely similar outward behavior based on published descriptions alone.

To address this problem, we have explored the use of machine vision approaches to quantitatively characterize and classify *C. elegans* uncoordinated mutants. In previous work, we built a tracking and imaging system that could follow and record an individual animal's movements over long time periods and save digital image data representing the animal's body posture over the course of the recording [6]. Algorithms were also devised to measure 94 features of a given mutant's body shape or locomotion pattern, making it possible to comprehensively test multiple aspects of behavior simultaneously. By using these features, it was possible to reliably distinguish examples of representative mutant types from one another using a binary decision tree algorithm (CART).

We also want to use these features to obtain a specific, quantitative definition of a particular mutant behavior pattern that would be diagnostic of a specific molecular defect and would facilitate quantitative comparisons between different mutant strains. In this study, we used image data collected by our automated tracking system to investigate the natural clustering of *C. elegans* behavioral data. From a complex data set consisting of 116 features measured from video recordings of 800 individuals representing 8 distinct genetic types, we used feature subselection and principal components analysis to represent each mutant type as a cloud of data points in low-dimensional feature space. We used k-means clustering and Euclidean distance measurements to explore the natural structure of the behavioral data and to compare the similarities of mutant patterns. These results therefore constitute a quantitative

characterization of the behavior of several important *C. elegans* mutant types, and demonstrate that mutants can be clustered using a complex behavioral signature based on quantitative image features.

2. IMAGE FEATURE EXTRACTION

Acquisition of image data: Routine culturing of *C. elegans* was performed as described in [4]. All worms analyzed in these experiments were young adults; fourth-stage larvae were picked the evening before the experiment and tracked the following morning after cultivation at 22°. Experimental animals were allowed to acclimate for 5 minutes before their behavior was analyzed. We used wild type worms and seven mutants (*unc-38*, *unc-29*, *goa-1*, *unc-36*, *unc-2*, *egl-19*, and *nic-1*).

C. elegans locomotion was tracked with a stereomicroscope mounted with a CCD video camera [6]. A computer-controlled tracker was used to maintain the worms in the center of the optical field of the stereomicroscope during observation. To record the locomotion of an animal, an image frame of the animal was snapped every 0.5 second for at least five minutes. Among those image pixels with values less than or equal to the average value minus three times the standard deviation, the largest connected component was found. The image was then trimmed to the smallest axis-aligned rectangle that contained this component, and saved as eight-bit grayscale data. The dimensions of each image, and the coordinates of the center of mass of the worm in the tracker field were also saved simultaneously as the references for the location of an animal in the tracker field at the corresponding time point when the images are snapped. The stereomicroscope was fixed to its largest magnification (50 X) during operation. Depending on the type and the posture of a worm, the number of pixels per trimmed image frame varied. The number of pixels per millimeter was fixed at 312.5 pixel/mm for all worms.

Image Pre-processing: To obtain the clean binary image, the background intensity level of the grayscale image was found first by taking the maximum of the values of the four corner points of the trimmed image (at least one of the corner points is always not part of the worm body). After finding the background level (b), a 5x5 moving window was scanned over the trimmed image, and the mean (m) and standard deviation (s) of the pixels inside the window were computed at every pixel position. If m was less than 0.7b or s was larger than 0.3m, then the pixel was considered to be a pixel of the worm body and was assigned a value 1. In order to clean up the spots inside the worm body, a morphological closing operator (binary dilation followed by erosion) was applied [7]. Next, the sequential algorithm for component labeling was used to remove unwanted isolated objects [8]. The connected components were labeled by scanning the image in x and y directions sequentially, and the largest

component was selected to guarantee that there will be only one object, the worm, in the binary image.

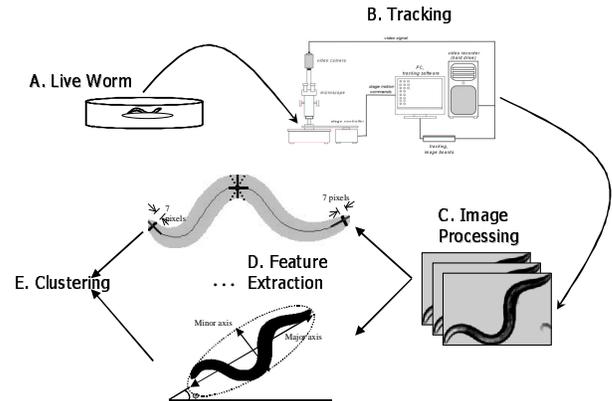


Fig 1. Flow chart of the data acquisition, image processing, and feature extraction steps. (A) Fourth-stage larvae were picked the evening before the experiment and tracked the following morning on a fresh plate. (B) A tracking system containing a high power microscope and a stage controller was used to track and record the worm locomotive information. A five-minute video sequence captured at 2Hz was stored in the PC hard drive for further processing. (C) Image processing steps removed noise and separated worm bodies from the background. (D) Feature extraction step extracted a total of 116 features from the binary image sequence. Some of the features measured the body size and posture of the worm, others measured movement. Eight genetic types with 100 five-minute recordings each were made. (E) The data were then fed through the k-means clustering algorithm for cluster analysis

Image Feature Extraction: All of the software for binarization, skeletonization, and feature extraction was coded in C and implemented on a UNIX machine. Some features (e.g., the area of the worm, that is, the number of pixels which make up the single binary object in the frame) could be computed on a single frame; these were computed for all 600 frames in the sequence. The average value, the maximum value and the minimum value were then computed for these 600 measurements. Some of the maximum and minimum values are outliers introduced by noise or errors during image capture and processing. To avoid using these extreme values, it was more useful to summarize the group statistics with such quantities as the 90th and 10th percentile values out of the population of 600 numbers. Other features could not be extracted from a single frame, for example, the movement between two frames, or the movement within 10 seconds (20 frames). Since there are approximately 600 frames total in a sequence, the movement between two frames could be computed 300 times if we take pairs of frames in a non-overlapping fashion, or it could be calculated 599 times taking pairs of frames in a sliding window or overlapping fashion. Likewise, for the movement within 20 frames, we could compute 581 values for overlapping 20-frame intervals. Quantities of this type were calculated in a sliding window fashion. As before, the average, max, min, and other order statistics can be computed from this set of numbers. A systematic view of the data acquisition and processing is shown in Figure 1.

The measured features included the minimum, maximum, and average values of the following: distance moved in 0.5, 5, 10, 15, 20, 25, and 30 seconds, number of reversals in 40 sec and 5 min, worm area, worm length, thickness at center and head/tail, ratio of these thickness to length, fatness, eccentricity and lengths of major/minor axis of best-fit ellipse, height and width of minimum enclosing rectangle (MER), ratio of MER width and height, ratio of worm area to MER area, angle change rate, and counts of the times that the worm coils or loops over its own body. We now describe in detail how several of these features are extracted from the image data.

Body skeleton/length: Obtaining the correct morphological skeleton for the binary worm object plays an important role in feature extraction. It serves as the basis for many body shape and posture related features. For example, the worm length can be readily calculated by counting the number of pixels on the skeleton. As shown in Figure 2, a skeletonizing algorithm described in [7] is applied to the binary image of the worm (Fig. 2c). Redundant pixels on the skeleton are eliminated by a thinning operation. Notice the track adjacent to the worm tail is mistakenly classified as a branch of the skeleton by the algorithm in this example. To prune down this skeleton, we first shrink the skeleton from all its end points simultaneously till only two end points are left. These two end points represent the longest end-to-end path on the skeleton. A clean skeleton (Fig. 2d) can then be obtained by growing out the two remaining end points along the unpruned skeleton by repeating a dilation operation.

Body thickness/fatness: The worm thickness was measured at the center and head/tail positions of the worm skeleton (the center position was the value at the center of the skeleton pixel list; the head/tail position was defined as the position which is 7 pixels away from each end of the worm body). In order to measure the center thickness, we first took a 9-pixel-long segment from the skeleton list, and computed the best fit line for the segment by a line fitting algorithm. Then we rotated the line by 90 degrees to get a perpendicular line to it. We traversed the perpendicular line in both directions from the center position until we reached the edges of the worm body, and then computed the distance between the two edges. We also rotated the perpendicular line by -5 and +5 degrees, and measured the thickness in those two directions. The minimum value of the three measurements was considered to be the center thickness. Similarly, in order to measure the head/tail thickness, we took two 9-pixel-long segments from each end of the skeleton list. After getting the best fit lines for the segments, we found the designated head/tail position by going back 7 pixels from the end of the worm body along the best fit line. Then we computed the thickness at these two measuring positions (one at each end) by traversing the perpendicular lines to the best fit lines. The minimum value of the two measurements was considered to be the

head/tail thickness. We also define the worm's fatness as the ratio of worm area to length.

Local movement: Many features characterize the global movement using the absolute distance traveled by the worm body centroid over various fixed time intervals. We also measured the relative offset of the centroid across the frames as an indication of the worm's local movement. This offset is defined as the movement of the centroid within the minimum enclosing rectangle (MER) from one frame to the next. The centroid location is normalized using the MER width and height, so the resulting coordinates are always between 0 and 1.

Angle change rate: The angle change, an important feature for distinguishing different worm types, is defined as the angle difference between any two consecutive 10-pixel skeleton segments along the skeleton [6]. A larger angle change rate means that a worm has deeper body bends. Figure 2(F-G) show typical skeletons from two different mutant types. The angle change rate is 15.51 for the skeleton in Fig. 2F compared with 8.45 in Fig. 2G.

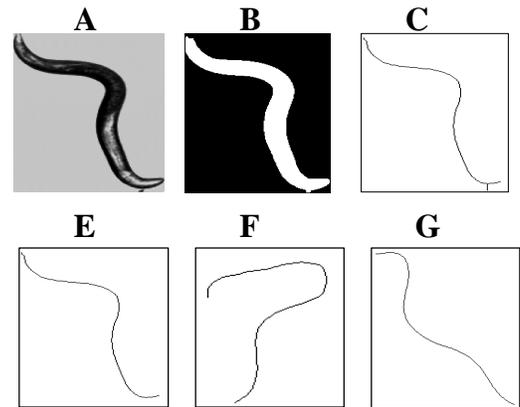


Fig 2. A-E illustrate the skeleton generating process. (A) Gray level image acquired from a video sequence containing the worm body and part of track. (B) Corresponding binary image after thresholding. (C) Skeleton after applying skeletonizing algorithm and redundant pixel removal. (D) Clean skeleton after pruning. E-G show typical skeletons from two different worm types. (F) typical *unc-2* skeleton. (G) typical *egl-19* skeleton.

3. NATURAL CLUSTERING

Feature selection: Since the inclusion of superfluous features during classifier training can lead to degradation in prediction accuracy [9], we wished to select a subset of the 116 initial features that were most representative of our data for use in cluster analysis. We screened the entire feature set using a backward elimination process based on the Lagrangian Support Vector Machine classifier [10,11], which was used because it generalizes well. The process starts from the full feature set. In each iteration, one feature is eliminated from the remaining set by evaluating all the possible subsets (n subsets, each containing n-1 features, need to be evaluated for an n-

element feature set) and selecting the subset that achieves the smallest training error rate as our next feature set. We use a low training error as an approximation of the importance of that feature. All the features can thus be ranked according to when they are eliminated in the backward elimination process. We repeat this process for all 8 mutant types in a pairwise fashion and generate 28 sequences of ranked features. We proceed with the features that appear at least five times among the top 10 features of each of 28 sequences and use these 18 features as our subset to represent the feature data. Parallel analyses were conducted using three scaling methods. We also compared the classification error of the first few principal components with feature subsets using these scaling methods. We concluded that the data were well represented using a subset of 18 features, with less than 4% cross-validation error rate (Fig. 3). These features included several measurements of speed averaged over different time periods, as well as the features described in Section 2.

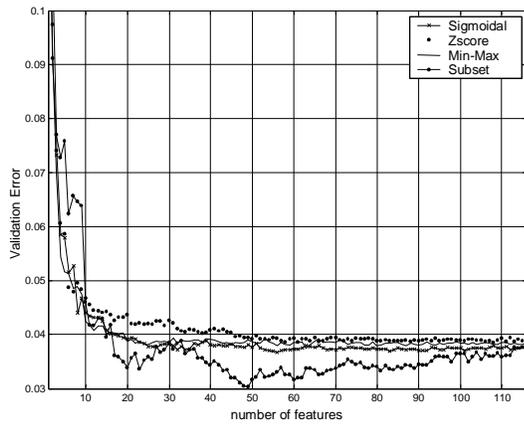


Fig 3. Comparison between different scaling methods and feature subset. The top three curves represent the 1 Nearest Neighbor (1-NN) classification error rate using Min-Max(scale feature value between -1 to 1), Zscore (normalize by standard deviation, and Sigmoidal (Zscore with sigmoidal scaling to limit the value to between -1 and 1) scaling, respectively [12]. The error was an average of 50 trials of 10-fold cross-validation result for each method. The features were selected from the first few Principal Components (PCs) of the entire 116 input features. All three scaling methods achieved similar performance, with the sigmoidal method slightly outperforming the other two. The fact that the error curves level off indicates most of the useful information for classification is heavily concentrated in the very first few PCs. The bottom curve shows the same cross-validation test but with a subset of features selected by a backward elimination method. The black curve also shows the adverse effect of increasing error rate with more features added. We opted to use the 18 most important features (see Feature Selection section).

K-means clustering and stopping rule: To further investigate the clustering of the data points, we applied the k-means clustering algorithm to find the natural clusters in the behavioral data. For this analysis, each data point was treated individually without regard to mutant type. We generated sufficiently many (10,000) random initializations for each k and tracked the error at the convergence to be reasonably confident that the global minimum was found. Figure 4 shows the cluster centers identified by the k-means algorithm; for each case, the

centers are marked by black squares. Although the actual k-means clustering was done using all 18 selected features, the data were visualized by showing the first two principal components. Table 1 shows the Euclidean distance between prototype centers (cluster centers).

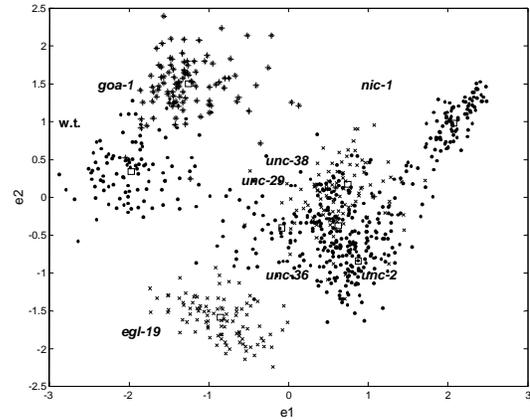


Fig 4. The plot shows all 800 data points represented in the space of their first two principal components using sigmoidal scaling. The data points from the same mutant type are marked by the same symbol.

	w.t.	goa-1	nic-1	unc-36	unc-38	unc-29	egl-19	unc-2
w.t.	-	1.6	4.1	2.9	2.7	2.1	2.3	3.1
goa-1		-	3.5	2.7	2.5	2.3	3.2	3.2
nic-1			-	2.2	1.6	2.6	3.9	2.3
unc-36				-	1.0	1.1	2.0	0.8
unc-38					-	1.0	2.4	1.1
unc-29						-	1.5	1.2
egl-19							-	2.0
unc-2								-

Table 1. Euclidean distance between prototype centers (cluster centers) measured in 18-dimension feature space. *wild* type and *nic-1* are the furthest apart. *unc-29* and *unc-38*, *unc-2* and *unc-36* are among the closest. This indicates a simple Euclidean distance in feature space can be used to quantify the relative similarity between mutant types.

A key issue in k-means clustering is to determine the optimal number of clusters for the data set. We used two algorithms to determine the optimal cluster number for our behavioral data: the gap statistic [13] and the information theoretic method [14]. The idea of the Gap Statistic is to standardize the graph of $\log(W_k)$ by comparing it to its expectation under an appropriate null reference distribution of the data. W_k is the total within-cluster sum of squares around the cluster centers, when there are k clusters. Since we have 800 points in our data set, the null reference distribution is generated by drawing 800 samples from a distribution that is uniform along each feature data dimension. This is repeated B times.

The expectation of the null reference $E\{\log(W_{kb}^*)\}$ can

be estimated as $1/B \sum_{b=1}^B \log(W_{kb}^*)$, where W_{kb}^* is the

within-cluster sum of squares of the b^{th} reference dataset, and B is the number of reference datasets. The distance between these two curves is defined as the Gap,

$$\text{Gap}(k) = 1/B \sum_{b=1}^B \log(W_{kb}^*) - \log(W_k), \text{ for } k=1, \dots, K,$$

where K is the maximum number of clusters defined by the user according to the expected range of clusters. We use a maximum of 10 centers ($K = 10$) and 5 reference datasets ($B = 5$). The sampling distribution can be measured by $s_k = sd_k \sqrt{1 + 1/B}$, where sd_k is the standard deviation of the reference null distribution. The formula to calculate the optimal number of clusters k_{opt} can be obtained as the first location where the gap curve starts to drop or level off. That is the first k that satisfies $\text{gap}(k) \geq \text{gap}(k+1) - a s_{k+1}$, where a is a multiplier adjusted to reject null mode. Here it is set to 2.

The Information Theoretic approach [14] tries to find the optimal number of clusters by fitting the within-cluster sum of square curve (distortion curve) with two hyperbolic curves breaking at the location of the optimal k . The location of the break can be measured in a transformed domain when applying a negative power to the distortion curves. The magnitude of the power is controlled by the dimensionality of the data. Here it is set to -4 . The transformed distortion curve usually can be approximated reasonably well by a piecewise linear function consisting of two straight lines with a break, or elbow, at the location of the optimal k . The optimal number of clusters can be easily obtained by finding the biggest jump, which is the difference between the successive points on the transformed distortion curve. The paper [14] provides theoretic justification and points out that this method can also provide suboptimal solutions by finding smaller jumps in the curve. This is particularly appealing given our objective of exploring the substructure of the data.

As shown in Figure 5, both methods identified 8 clusters as the optimal number, with each cluster composed primarily of a single mutant type (Table 2). The information theoretic approach identified an additional suboptimal solution of 6 clusters (Fig. 5 and Table 2). In this suboptimal classification, the calcium channel mutants *unc-36* and *unc-2* were grouped into a single cluster and the nicotinic receptor mutants *unc-29* and *unc-38* into another cluster. Together, these results demonstrated that worms of the same mutant type tend to exhibit similar behavioral patterns and further showed that cluster analysis can be used to assess phenotypic similarities between different mutant classes.

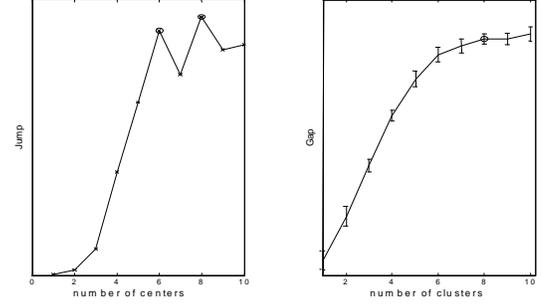


Fig 5. Natural clustering results. Jump plot (left) by information theoretic method. The optimal and suboptimal number of clusters, marked by circles, were identified as the most and second most significant peaks Gap plot (right) by gap statistic method. The optimal number of clusters, marked by a circle, was identified as the gap curve first started to level off.

Center	#1	#2	#3	#4	#5	#6	#7	#8
w.t.	92	5	0	0	0	3	0	0
goa-1	4	93	0	2	1	0	0	0
nic-1	0	0	96	1	2	0	0	1
unc-36	0	0	0	64	16	3	0	17
unc-38	0	0	1	3	72	17	0	7
unc-29	3	0	1	10	18	54	3	11
egl-19	0	0	0	2	0	0	98	0
unc-2	0	0	0	14	5	1	0	80

Center	#1	#2	#3	#4	#5	#6
w.t.	93	5	0	0	2	0
goa-1	4	93	0	2	1	0
nic-1	0	0	98	2	0	0
unc-36	0	0	0	87	22	0
unc-38	0	0	2	9	89	0
unc-29	3	0	3	27	54	13
egl-19	0	0	0	2	0	98
unc-2	0	0	0	94	4	2

Table 2. The 800 data points were classified into 8 clusters (upper, optimal number of clusters) or 6 clusters (lower, suboptimal number of clusters) based on their shortest distance to the cluster centers identified by the k-means algorithm.

4. CONCLUSION

Quantitative definition of behavioral mutant types:

We have shown here that quantitative morphological and locomotion features obtained from digital video recordings can be used to define the behavioral types of *C. elegans* mutants. For example, *unc-36*, *unc-29*, *unc-38* and *unc-2* have all been categorized as "weak kinkers", a term that has been difficult to define precisely. From Table 1, it is apparent that these mutants share many common effects on the variables used in our

classification; in particular, all have a substantially higher angle change rate and substantially lower centroid movement and global speed parameters than wild-type. This combination of characters (increased body bending and a decreased rate of movement) thus provides an operational definition of the "kinker" type. Likewise, the combination of increased centroid movement and increased angle change rate provides a functional definition of *goa-1*'s "hyperactive loopy" type, while increased length and length/eccentricity and decreased angle change rate and speed define the "long, slow and floppy" *egl-19*. Thus, it has been possible not only to obtain precise quantitative descriptions of mutant behavioral types whose definitions had previously been subjective and qualitative, but also to resolve subtle differences within broad classes such as kinker Uncs.

Prospects for using behavioral types for bioinformatic analysis: The application of machine-based pattern recognition methods also allowed us to probe the similarities between different behavioral patterns based on their clustering in multidimensional feature space. In general, the pattern of clustering mirrored the known similarities in molecular function and cellular site of action of the mutant gene products. For example, the *unc-2* and *unc-36* mutants, which are defective in the α -1 and α -2 subunits respectively of the neuronal N-type calcium channel, formed a single cluster in the sub-optimal k-means clustering (Figure 3d), and the centers of these two types' data clouds were the closest together by Euclidean distance. Likewise, *unc-29* and *unc-38*, which respectively encode α and β nicotinic receptor subunits with overlapping expression patterns, formed a single cluster in the sub-optimal clustering and had centers that were relatively close in feature space. In fact, the centers for all four of these types (which have all been designated as kinker Uncs and all encode excitatory ion channels whose focus of action is primarily at body muscle neuromuscular junctions) were closer to one another than to the other Unc mutants or wild-type. Thus, the quantitative behavioral signature obtained through image feature extraction appeared to correspond well to the underlying functional defects of the mutants we analyzed.

We anticipate that this type of comprehensive quantification of mutant behavioral types will have powerful applications in functional genomic studies. Clustering and pattern recognition analysis of microarray-derived gene expression profiles has provided important information about the likely functions of novel gene products in *C. elegans* and other organisms [15]. In principle, a behavioral type represents a similarly complex quantitative signature whose direct linkage to nervous system activity makes it particularly useful for classifying genes that function in excitable cells. In several genome-wide deletion and RNAi-based knockout surveys undertaken in *C. elegans*, the identification and classification of behavioral types has been a crucial limiting factor [16,17]. Using the machine-based approaches described here, it should be possible to record

the behavior of an uncharacterized knockout strain, compare its behavioral pattern to a database of known mutants, and make an informed initial hypothesis about the molecular pathways in which the mutant gene product participates.

REFERENCES

- [1] J. Sulston and H. Horvitz, 1977 Post-embryonic cell lineages of the nematode *Caenorhabditis elegans*. *Dev. Biol.* 56, 1977, 110-156.
- [2] J. Sulston, E. Schierenberg, J. White, and J. Thomson, 1983 The embryonic cell lineage of the nematode *Caenorhabditis elegans*. *Dev. Biol.* 100, 1983, 64-119.
- [3] J. White, E. Southgate, N. Thomson, and S. Brenner 1986 The structure of the *Caenorhabditis elegans* nervous system. *Philos. Trans. R. Soc. Lond. (Biol.)* 314, 1986, 1-340.
- [4] S. Brenner, The genetics of *Caenorhabditis elegans*. *Genetics*, 77, 1974, 77-94.
- [5] J. Hodgkin, Male phenotypes and mating efficiency in *Caenorhabditis elegans*, *J. Genetics* 103, 1983, 43-64.
- [6] J. Baek, P. Cosman, Z. Feng, J. Silver, and W.R. Schafer, Using machine vision to analyze and classify *C. elegans* behavioral phenotypes quantitatively. *J. Neurosci Meth*, 118, 2002, 9-21.
- [7] R. Gonzalez and R. Woods, *Digital Image Processing*, (Prentice Hall, New Jersey, 2002, Second Edition)
- [8] R. Jain, K. Rangachar, and B. Schunck, *Machine Vision*, (New York, McGraw-Hill, 1995)
- [9] R. Kohavi and G. John, Wrappers for features subset selection, *Artificial Intelligence*, Vol.97, No.1-2, 1998, 161-177.
- [10] O. Mangasarian and R. Musicant, Lagrangian Support Vector Machines, *J. Machine Learning Research*, 1, 2001, 161-177.
- [11] F. Model, P. Adorjan, A. Olek, and C. Piepenbrock, Feature Selection for DNA methylation based cancer classification. *Bioinformatics*. 17 Suppl, S1, 2001, 57-64.
- [12] D. Grossman, *Short Course in Data Warehousing and Data Mining*, (online material http://www.ir.iit.edu/~dagr/DataMiningCourse/Spring2001/Notes/Data_Preprocessing.pdf), 2002.
- [13] R. Tibshirani, G. Walther, and T. Hastie, Estimating the number of clusters in a dataset via the Gap statistic, *J. Royal Statistical Society Series B.* 63, 2001, 411-423.
- [14] C. Sugar and G. James, Finding the number of clusters in a data set: An information theoretic approach. *J. American Statistical Assoc.* 2003 To appear
- [15] S. Kim, D. Poole, L. Waggoner, A. Kempf, D. Ramirez, A. Treschow, W. R. Schafer, Genes affecting the activity of nicotinic receptors involved in *C. elegans* egg-laying behavior, *Genetics*, 157, 2001, 1599-1610.
- [16] A. Fraser, et al., Functional genomic analysis of *C. elegans* chromosome I by systematic RNA interference. *Nature*, 408, 2000, 325-330.
- [17] P. Zipperlen, A. Fraser, R. Kamath, M. Martinez-Campos and J. Ahringer 2001 Roles for 147 embryonic lethal genes on *C. elegans* chromosome I identified by RNA interference and video microscopy *EMBO J.* 20, 2001, 3984-3992.