Sequential Movement Pattern Discovery of Caenorhabditis elegans using Hidden Markov Models

Anthony Gualandri, Sarah Hunt, Scott Page, Steven Rotter  
*CSC 672 - Winter 2018*

# **1. Abstract**

Caenorhabditis elegans, or C. elegans, roundworms have been an interest to neuroscientists due to their similarities in genetic makeup compared to humans. Our research is intended to provide biologists and neuroscientists with a way to differentiate between C. elegans Wildtypes and modified Mutant types. If we can identify a distinct yet shared set of behavioral patterns between worms at the multi-second level of movement through the use of Hidden Markov Modeling (HMM), then perhaps we can identify how mutated worm behavior differs from Wildtype worm behavior. By clustering the position of the worm over a series of time, we were able to identify four movement patterns (Ahead, Ahead Slow, Right and Left) to differentiate the Wildtype fed, Wildtype unfed, and Mutated fed worms. We were also able to discover three hidden states through our Hidden Markov Modeling. Results showed that the Wildtype fed worm (N2\_f1) typically stayed in a state of forward movement at full speed and rarely turned. Wildtype unfed worms (N2\_Nf4 and N2\_Nf5) had a tendency to turn more frequently, with the N2\_nf5 worm showing bias toward left turns. The mutant worm (tph1\_f6) captures both fed and unfed Wildtype behavior, as it shows a bias toward right hand turns but for the most part moves forward in a smooth path like the Wildtype fed worm. Future work in the area of image processing can be done in order to apply nonparametric approaches to C. elegans data.

# **2. Introduction**

C. elegans worms have been studied for decades because of their genomic simplicity. In 1986, scientists were able to create a connectome, or map of all 302 neurons for C. elegans and a wiring map with all the connections between each neuron [1]. Researchers are interested in studying C. elegans behavior in hopes of understanding how certain neurons in the human brain affect certain behaviors, as the genome makeup of these worms are similar to humans (40% homologous) [2]. If researchers can identify which neurons trace back to certain behaviors in C. elegans, then they can potentially explain certain behaviors in humans using a similar mapping.

Our research is intended to provide biologists and neuroscientists with a way to differentiate between C. elegans mutant types through a better understanding of what characterizes their sequential movement and posture-based behaviors. Due to the small size and speed of worm behavior, scientists aren't completely sure whether the individual behavioral “states” observed through visual inspection are the correct or complete set of behaviors. If we can identify a distinct yet shared set of behavioral patterns between worms at various time granularities, then we can identify how mutated worm behavior differs from non-mutated worm behavior.

2.1 Related Work

Previous C. elegans research attempting to differentiate between a certain set of well understood behavioral states (local exploitation, long distance exploration, and immobility due to satiation) points towards a potentially continuous set of hidden states in between these three as a result of their use of Hidden Markov Modeling [3]. Other research seeking to explain meaningful behavioral differences in swimming patterns of C. elegans placed in treated and untreated solutions found discernable differences, also through the use of Hidden Markov Modeling [4].

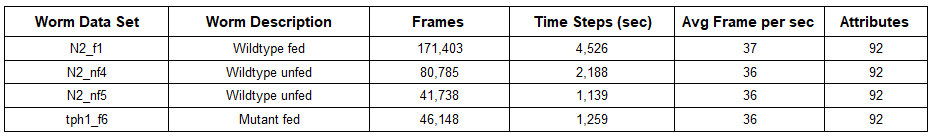
Related research in recognizing organized sets of time sequential behaviors through the use of Hidden Markov Modeling has been an area of investigation for the better part of two decades. Early research focused on recognizing human action concentrated on domains where the specific set of differentiable behaviors were fully known [13]. More recent research has sought to quantify such activities where the set of behaviors is entirely unknown and it is a direct goal of the research to discover such information. [8][14][15] have developed a recent class of models to effectively accomplished this on a number of different yet related sets of time series such as the dance of honey bees, stock price volatility, speaker diaritization, and human exercise motions, demonstrating the powerful generalizability of such methods. [5] uses this same class of models with mice for the express purpose of understanding better neurological functioning. It was discovered that mice reuse a set of stereotyped behaviors that are organized at the sub-second level and have defined and predictable transition probabilities.

# **3. Materials and Methods**

Data was collected using a table with a mobile camera and used machine vision techniques [11] from four different worms of various types: Wildtype with food (N2\_f1), Wildtype without food (N2\_nf4 and N2\_nf5), and fed but genetically modified to be without a sense of satiation (tph1\_f6). Each data set has 92 features that range from posture, speed, size, shape, and movement. Features Centroid X and Centroid Y included in the data set captured the position of the worm.

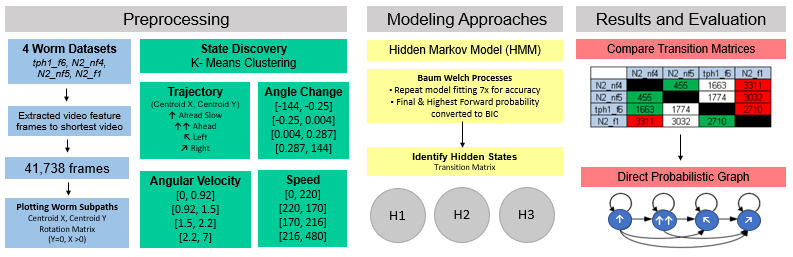
For consistency, we set the number of rows of data to use for each worm type at the first ‘n’ rows with ‘n’ equal to the minimum number of rows provided for each worm type. By doing this we compare the same length of time for each worm. Wildtype N2\_nf5 was the shortest file with 41,738 rows (roughly 19 minutes) of data. Therefore, only the first 19 minutes from each data set were used.

The average frame rate per second varied slightly between worm videos. N2\_f1 worm had a higher frame rate per second (37 vs. 36 for other worm types). This was due to a bug in the mobile camera code, that was fixed in subsequent worm videos. It was determined that this one frame per second difference would not greatly impact the results of this research.

 *TBL 1. Dataset Description*

3.1. Methodology

Our proposed methodology seeks to discover hidden states by mapping a set of downsampled time steps to a varying number of k states of a HMM using the Centroid X and Centroid Y features. Speed, Angle Change, and Angular Velocity features were also clustered together to create a single categorical variable as input to the model.



*FIG 1. Methodology flowchart.*

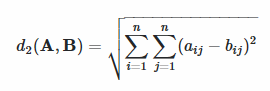
3.1.1. Data Preprocessing

#### 3.1.1.1. Data Discretization

Using the results from [10] and other features that describe movement, three base features were chosen from the datasets to perform a comparative analysis on different Markov chain transition matrices: Speed, Angle, and Angular Velocity. Angle was not used by itself but was used to calculate a new feature, Angle-Change, (the difference between measurements of two sequential frames) to get a better measurement for movement that is not based on the worm’s random orientation on the agar plate. Two additional features were created using the Angle-Change feature by grouping the labelled values into sequences of two and three. As an example, if the first six Angle-Changes were labelled as “A”, “A”, “B”, “B”, “B”, “C” then the new feature would be three values with sequence length two: “AA”, “BB”, “BC”. With the sequence length three there would be two values: “AAB”, “BBC”. One additional feature was created by combining two features, Speed and Angle-Change, to come up with a single feature to describe speed, turning direction and turning sharpness.

The features in the dataset are continuous variables, but to calculate a transition matrix we discretized them into three or four bins based on the feature. As an example, the change in angle would be discretized into three bins to describe a left turn, right turn, or a straight-ahead movement. The breakpoints of the bins were calculated to keep the bins evenly distributed for the four worms in aggregate. This was done by taking the first ‘x’ values from each worm, pooling them together, and then discretized these pooled values to keep the bins evenly distributed. This method for discretization was chosen to help reveal similarities and differences in the worms while keeping outliers at a minimal that would skew the calculation of distances between the worms.

For each feature a Markov Chain transition matrix was calculated for each worm. The matrices use the actual instance count for each worm type to move from one bin state to another; not the probability of moving from one bin state to another. Using the count also helped minimize potential over calculating the distance calculation.

The Euclidean distance calculation; the square root of the sum of the squares of the difference between matching cells was used to calculate the distance between two matrices (Equation 1).

Equation 1

#### 3.1.1.2. Worm Sub-Paths

Each worm’s path denoted by the time series of Centroid X and Centroid Y was broken into equal sub-paths of 50, 1/10 downsampled frames (originally 500 frames downsample to 50 frames). Worm location on the agar plate and direction of travel was assumed to be unimportant when comparing worms since there was no known reason for worms to prefer any particular region of the plate, and starting direction of worm was not controlled in data collection experiments making long term path of worms incomparable. Due to this, each worm sub-path was adjusted to begin travel at position (0,0). Worm’s sub-path was then rotated via rotation matrix such that the final sub-path position would fall along y = 0 and x > 0 point. Each rotated sub-path forms a trajectory since it contains both position information as well as time. This new trajectory axis space contains three axis: rotated\_xpos, rotated\_ypos, and time. Rotated\_xpos represents the axis of farthest travel between the start and end points of the worm’s sub-path, rotated\_ypos represents the orthogonal secondary axis of worm travel, and time represents the downsampled frame number of the sub-path starting at 0 and increasing to 49.

3.1.2. Observed Trajectory Cluster Discovery

The rotated sub-paths of all worms were aggregated for clustering. Since each worm contributed the same number of rotated sub-paths to the clustering dataset, clustering bias issues were limited. Clustering was performed using a variant of k-means designed for longitudinal and trajectory data called KML3D, with runtime complexity of O(n\_tradj \* # of clusters \* rerolls \* tradj\_length) [12]. The temporal values of each rotated sub-path were denoted by downsampled frame numbers adjusted to 0 for the first frame of the rotated sub-path, and incrementing by one up to the final downsampled frame (50) within each rotated sub-path. Cluster counts from three to seven were inspected for quality. Individual trajectory cluster membership was computed using Euclidean distance from cluster means.

#### 3.1.2.1. Observed Trajectory Cluster Evaluation

Cluster quality was measured using the Calinski Harabatz chart provided by the R KML3D package [FIG 1]. Calinski Harabatz score of each rerolling of initial cluster centroids identifies quality of clusters as (Separation) / (Compactness) by dividing average between cluster sum of squares [17]. The cluster count that produced the highest Calinski Harabatz score over the majority of rerolls was selected as the optimal cluster count.

To characterize cluster meaning, the mean values of the clusters rotated\_xpos and rotated\_ypos were plotted relative to time [FIG 1]. Additionally, the identified sequence of trajectory states was plotted on the original Centroid X and Y axis [FIG 2]. Characterizing notes were taken to describe what the worm was performing in each cluster [TBL 2] and summary statistics of worm trajectory states were computed [TBL3.]

3.1.3. Markov Decision Process Modeling

Each worm’s optimal observed sequence of KML3D trajectory clusters was fitted with Hidden Markov Models (HMM) using the Baum Welch algorithm with runtime complexity of O(n\_states^2 \* sequence\_length). The number of hidden states was varied between three to ten, and fitting was repeated seven times for each set of hidden states to limit impact of local minimum via bagging. HMM fitting was performed in parallel to reduce training times and support exploration.

3.1.3.1. Markov Decision Process Evaluation

The HMM’s were evaluated for fit of their respective worm’s data by computing the forward probability of each model with respect to its training data. Forward probability measures the likelihood that an HMM produced an observed sequence of states. Since many models were trained for each HMM’s hidden state count, the maximum forward probability of each bagged set of models was used for evaluation. These maximum forward probabilities were converted to BIC values to further support comparison. The hidden state count that generally produced the lowest training data BIC values was selected as the optimal number of hidden states.

After the number of hidden states was established, graphical representations of the trained HMM’s were used to identify similarities and differences in worm decision processes. Each graphical HMM represents hidden states with green nodes, and emission states with red nodes. Transition probabilities are noted with green curved edges and emission probabilities by red curved edges. Thicker edges correspond to more common transitions & emissions probabilities. Edges representing less than 1% chance have been filtered out. Transition and emission edges are read clockwis*e and* all graphical HMM’s are laid out using the same force atlas 2 algorithm in gephi with avoid overlap enabled. This makes comparison between HMM’s more graphically meaningful.

To supplement graphical HMM findings, forward probabilities of models were also computed with respect to other worms observed trajectory state sequences. These were converted to BIC and a table was created for inspection [TBL 4] BIC results are only comparable between models on the same sequence of observed trajectory states, so rescaling was necessary to identify patterns between trajectory sequences. This rescaling process involved two steps:

1. The lowest value of BIC for each sequence of observed states was divided through the other models BIC’s with respect to the sequence of observed states; resulting in a percentage of additional error that each worm’s HMM contained relative to the best model for a given sequence of states.
2. All values except the self HMM & state combination were then negated and rescaled from -1 to 1 with the highest additional error receiving a score of -1 and the lowest additional error receiving a score of 1.

These rescaled values were plotted using the corrplot library, where darker blue represents better HMM state sequence fits, and darker red represents worse HMM state sequence fits [**FIG10**].

# **4. Results**

4.1. Transition Matrices

With worms N2\_nf4 and N2\_nf5, being the same worm type, we expected the distances between the two to be the closest when comparing each of these worms against the other three. We also expected worm tph1\_f6, being insatiable, to be closer to N2\_nf4 and N2\_nf5 as it wants to search for food.

The Speed variable had N2\_nf4 measuring furthest away from N2\_nf5 than from the other two worms while N2\_nf5 measured closest to N2\_nf4. N2\_nf5 exhibited very different speeds than the rest as it was always the furthest in distance away from each other worm. The opposite of what was expected occurred between tph1\_f6 and N2\_f1. The tph1\_f6 worm was closest to N2\_f1. N2\_nf1 was closer to tph1\_f6 than the other two worms [TBL5, TBL8].

Angular Velocity and all three Angle-Change (sequence lengths 1, 2, and 3) all had the same results. N2\_nf4 measured closest to N2\_nf5 and vice-versa. Tph1\_f6 was furthest from N2\_f1 but the opposite was not true. N2\_f1 was closest to tph1\_f6 [TBL 6, TBL 7, TBL 8].

The combination variable Speed & Angle-Change also measured N2\_nf4 closest to N2\_nf5 and vice-versa. Both Tph1\_f6 and N2\_f1 were furthest away from N2\_nf5. N2\_f1 measured closest to tph1\_f6 {TBL, TBL 9].

4.2. Observed Trajectory Cluster Results

In general, it was discovered that macro worm movement’s can be classified into three to four trajectory clusters using KML3D [FIG 1]. The primary three clusters representing forward motion, left turns, and right turns. When additional clusters are added, worm trajectory further subsets into more and less rapid forward motion before additional turning states are added. This study emphasized longer trajectory lengths best fit with the following 4 clusters: Ahead, Ahead Slow, Right, and Left. These clusters are characterized in detail in [TBL 2] and the cluster centroids plotted in [FIG 1]. Further visual confirmation of these cluster descriptions is provided in [FIG 2].

Worm trajectory tendencies are summarized in [TBL 3]. It was observed that Wildtype fed worm N2\_f1 spent the most time of all worms moving ahead. All other worms spent more time turning than N2\_f1 suggesting a link between satiation and turning. Overall, Wildtype unfed worm N2\_nf5 was the only worm to turn left more than right. Due to turning bias differences between worms N2\_nf4 and N2\_nf5 (who are both of the same type), it is possible that worms have individual preferences when it comes to turning bias. The insatiable mutant worm with food tph1\_f6 presented with behaviors that appear to blend N2\_f1’s forward movement bias with N2\_nf4’s right turn bias suggesting that while the worm is not full, its need to search for food via turning is regulated by the prevalence of food in its environment.

On visual inspection of each of the four worm’s paths, significant differences between fed and unfed worms were observed. In general, fed worm’s paths in [FIG2] and [FIG 8] maintain smooth paths with limited abrupt changes in direction. This is contrary to the paths in [FIG 4] and [FIG 6] that contain few smooth sections and numerous abrupt direction changes. This further suggests that worms that are able to find food in abundance prefer smooth locomotive paths, where worms in search of food, frequently deviate from a smooth path. This may indicate that Ahead and Ahead Slow locomotion requires the least amount of energy, and is the preferred form of motion of worms.

4.3. Markov Decision Process Results

After many trials of varying trajectory clusters and hidden state counts, three hidden states were chosen as the optimal balance between model fit (low BIC) and model scrutability (low number of hidden states).

4.3.1. Graphical HMM Results

It was noted that [FIG 3] turning emission states are only weakly connected to the hidden decision states, additionally no hidden states share left and right turning behavior. These two observations correspond with the tendency of N2\_f1 to move ahead at full speed more than the other worms and perform the least amount of turning behaviors, as observed in N2\_nf4 and N2\_nf5.

N2\_nf4’s [FIG 5] turning emission states are more tightly connected to hidden states than N2\_f1, identifying increased turning behavior. Since both Right and Ahead Slow clusters share the repeated H1 state, N2\_nf4 tends to produce tight right turning clusters. Additionally, hidden state number two can produce all emission states, resulting in occasional random behavior. This random behavior can be described as “jagged”, or not very smoothly connected cluster segments, when plotted in [FIG 4] and is believed to indicate N2\_nf4 is searching for food.

[FIG 7] contains significant hidden state circulation H1 🡪 H3 🡪 H2 🡪 H1 and repetition. This suggests N2\_nf5 tends to produce repeating trajectory patterns consisting of repeated Ahead Slow followed by Left and Ahead trajectories, and then a final rare Right. This strong cyclic transition of hidden states and left turn bias is unique to N2\_nf5. Since both Left and Ahead emissions are strongly produced from both hidden state number two and hidden state number three, the worms path contains wider food search clusters than N2\_nf4, and they tend to be left turn dominated. Since hidden state 2 connects to both left and right turning states, the path of N2\_nf5 can occasionally become “jagged” as shown in [FIG 6] this may be an indication that N2\_nf5 is searching for food.

[FIG 9] contains attributes of all three other worm models: (1) Similar to N2\_f1 the Ahead and Ahead Slow emission states are tightly connected to the hidden states and no hidden states strongly share both right and left emissions resulting in smooth path. (2) Similar to N2\_nf4, the right turning emission state is tightly connected to a repeating hidden state, indicating a significant bias to making right turns. (3) Similar to N2\_nf5 all hidden states strongly repeat themselves.

However, unlike all other worms graphical HMM’s, tph1\_f6’s hidden states have relatively poor interconnection (low between state transition probabilities). Specifically, hidden state 2 represents an island hidden state that is rarely transitioned to or from. This explains [FIG 8] that begins relatively straight, then transitions into an endless right turn, and finally resumes relatively straight motion again. It seems that even though tph1\_f6 can not be satiated due to the mutation, it doesn’t have to aggressively search for food, since food is plentiful on its plate, limiting sporadic turning behavior.

4.3.2. Model Fit Results

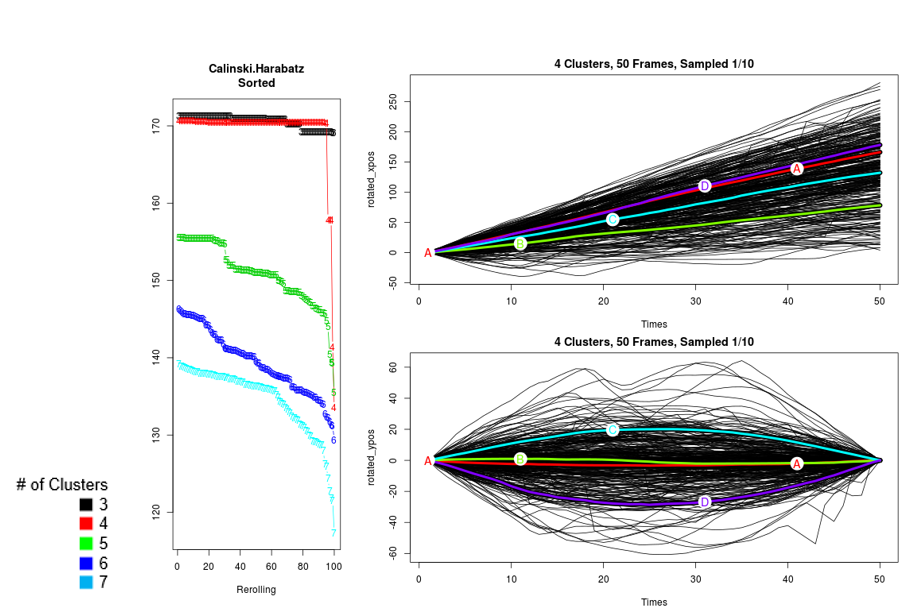
While graphical comparison of HMM structure can prove insightful and complementary to worm path inspection, forward probability can also be used to identify the effectiveness of an HMM’s fit to data. As described in [3.1.3.1. Markov Decision Process Evaluation](#_uq7lgodoxne) the forward BIC of each worm model was computed with respect to each other worms observed trajectory sequence, and then rescaled for comparison. On inspection of [FIG 10] it is evident that the mutant worm’s (tph1\_f6) decision process provides a better overall explanation of the other worms observed trajectory behaviors than any other study worm (represented by multiple dark blue circles in the bottom most row of plot). This is supported by the observed similarities between tph1\_f6’s graphical model and the other worms.

Interestingly, this contrasts with the relatively high error that N2\_f1 and N2\_nf5 worm models experienced when trying to explain tph1\_f6’s observed sequence of trajectory states. This suggests that while tph1\_f6’s decision process contains aspects of the other worms, its observed sequence of trajectory states is not easily explained by N2\_f1 and N2\_nf5’s decision models. Most likely this is due to the relative sparsity between hidden states in tph1\_f6’s model.

Additionally, N2\_nf4 and N2\_nf5 both received strong marks with respect to their models ability to fit the other’s sequence of observed states. This seems reasonable considering the two worms are both wild and unfed. However, this is an interesting finding considering the significant graphical HMM differences between the HMMs and worm turning bias differences. This suggests that the additional layer of abstraction an HMM provides may be critical when classifying an unknown worm, because of the HMM’s ability to overlook fundamental observed differences such as turning bias.

**5. Tables and Graphs**

5.1. Trajectory Clustering



**FIG 1.** *KML3D Clusters. Calinski Harabatz Sorted plot representing cluster quality with target numbers of clusters from 3 to 7. Rotated\_xpos vs time and rotated\_ypos vs time plots represent individual worm trajectories as well as optimal 4 cluster centroids. Thick colored lines are representative of rotated trajectory cluster centroids. Thin black lines are representative of individual rotated tradjectories clustered. Individual trajectories cluster membership is determined by euclidian distance from cluster centroids.*

### 

***TBL 2.*** *Cluster Descriptions*

### 5.1.1. Time per Observed Trajectory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Worm Name** | **Ahead** | **Ahead Slow** | **Right** | **Left** |
| **Wildtype fed (N2\_f1)** | 45% | 34% | 13% | 8% |
| **Wildtype unfed (N2\_nf4)** | 35% | 27% | 24% | 14% |
| **Wildtype unfed (N2\_nf5)** | 31% | 22% | 13% | 34% |
| **Mutant fed (tph1\_f6)** | 28% | 30% | 37% | 5% |

***TBL 3.*** *Percentage of time spent per observed trajectory.*

5.1.2. Graphical HMM Plotting Notes

Each graphical HMM represents hidden states with green nodes, and emision states with red nodes. Transition probabilities are noted with green curved edges and emission probabilities by red curved edges. Thicker edges corresponding to more common transitions & emissions probabilities. Edges representing less than 1% chance have been filtered out. Transition and emission edges are read clockwis*e such that in [FIG 3] H3* Transitions to H1 far more frequently than it transitions to H2. All graphical HMM’s are laid out using the same force atlas 2 algorithm in gephi with avoid overlap enabled. This makes comparison between HMM’s more graphically meaningful.

5.1.3. Wild Fed Worm N2\_f1

|  |  |
| --- | --- |
| **FIG 2.** *N2\_f1 Observed Trajectory States. See [TBL 2] for details.* | ***FIG 3.*** *N2\_f1 Graphical HMM. Red emission states, green hidden states, edges read along clockwise curvature, edge thickness denotes probability. For more details please see* [*5.1.2. Graphical HMM Plotting Notes*](#_tja90ynv50r5) |

5.1.4. Wild Unfed Worm N2\_nf4

|  |  |
| --- | --- |
| ***FIG 4****. N2\_nf4 Observed Trajectory States. See [TBL 2] for details.* | ***FIG 5****. N2\_nf4 Graphical HMM. Red emission states, green hidden states, edges read along clockwise curvature, edge thickness denotes probability. For more details please see* [*5.1.2. Graphical HMM Plotting Notes*](#_tja90ynv50r5) |

5.1.5. Wild Unfed Worm N2\_nf5

|  |  |
| --- | --- |
| ***FIG 6.*** *N2\_nf5 Observed Trajectory States. See [TBL 2] for details.* | ***FIG 7.*** *N2\_nf5 Graphical HMM. Red emission states, green hidden states, edges read along clockwise curvature, edge thickness denotes probability. For more details please see* [*5.1.2. Graphical HMM Plotting Notes*](#_tja90ynv50r5) |

5.1.6. Mutant Fed Worm tph1\_f6

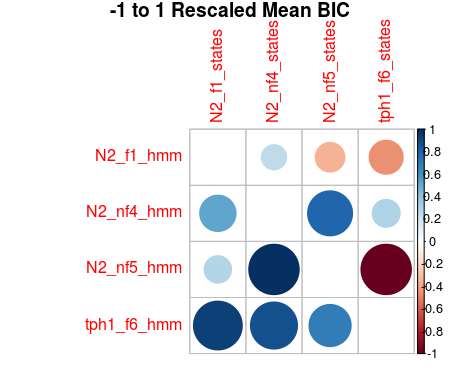
|  |  |
| --- | --- |
| ***FIG 8.*** *tph1\_f6 Observed Trajectory States. See [TBL 2] for details.* | ***FIG 9.*** *tph1\_f6 Graphical HMM. Red emission states, green hidden states, edges read along clockwise curvature, edge thickness denotes probability. For more details please see* [*5.1.2. Graphical HMM Plotting Notes*](#_tja90ynv50r5) |

## 

5.1.7. Cross Worm Trajectory HMM Fit Findings

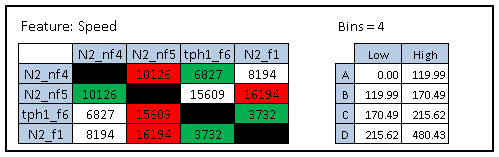
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Worm Model** | **N2\_f1 States** | **N2\_nf4 States** | **N2\_nf5 States** | **tph1\_f6 States** |
| **N2\_f1 HMM** | 183 | 301 | 344 | 262 |
| **N2\_nf4 HMM** | 239 | 218 | 278 | 229 |
| **N2\_nf5 HMM** | 249 | 259 | 225 | 285 |
| **tph1\_f6 HMM** | 220 | 267 | 284 | 168 |

***TBL 4.*** *Forward BIC Results. Rows represent specific worm models using 3 hidden states, columns represent specific worm sequences of observed trajectory states. Observed trajectory states discovered in [FIG 1] and described in [TBL 2].*

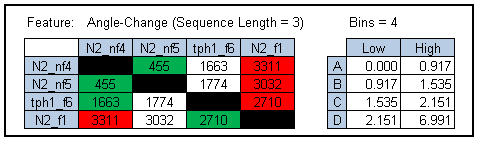


***FIG 10.*** *Rescaled Forward BIC Results. BIC scores present in [TBL 4] were rescaled and inverted to allow for comparison across sets of states & models. Scores along diagonal omitted. Darker blue circles have limited additional error vs best models, Darker red circles have maximal additional error vs best models.*

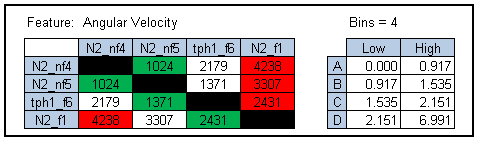
5.2. Transition Matrices Distance Measurements



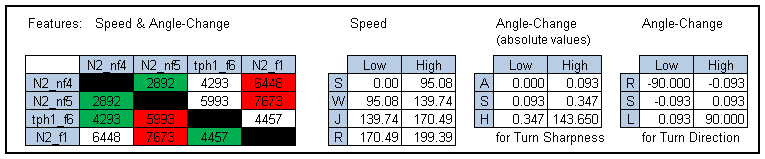
**TBL5.** Speed Distance Measurements. Values represent the Euclidean distance (Equation 1) between two worms’ transition count matrices. Green/Red highlighting represent nearest/furthest distances from one worm (vertical column) to the other three worm types (horizontal row). Bins represent the discretization breakpoints for re-labeling.



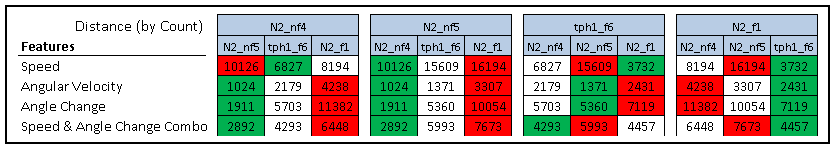
**TBL 6:** Angle-Change Distance Measurements. Values represent the Euclidean distance (Equation 1) between two worms’ transition count matrices. Green/Red highlighting represent nearest/furthest distances from one worm (vertical column) to the other three worm types (horizontal row). Bins represent the discretization breakpoints for re-labeling.



**TBL 7:** Angular Velocity Distance Measurements. Values represent the Euclidean distance (Equation 1) between two worms’ transition count matrices. Green/Red highlighting represent nearest/furthest distances from one worm (vertical column) to the other three worm types (horizontal row). Bins represent the discretization breakpoints for re-labeling.



**TBL 8:** Speed & Angle-Change Distance Measurements. Values represent the Euclidean distance (Equation 1) between two worms’ transition count matrices. Green/Red highlighting represent nearest/furthest distances from one worm (vertical column) to the other three worm types (horizontal row.) Bins represent the discretization breakpoints for re-labeling.



**TBL9:** Side-by-side comparison of the tables 5-8.

# **6. Discussion**

6.1 Summary of Results

|  |  |
| --- | --- |
| **Feature** | **Key Takeaway** |
| Trajectory | * Mutant worm decision process captures both fed and unfed worm behavior * Wildtype fed worms move ahead more than Wildtype unfed worms * Wildtype unfed worms have a tendency to turn more frequently * Wildtype unfed worm 5’s decision process doesn’t explain Mutant worm behavior * Wildtype unfed worms 4 and 5 have different turning biases (right vs left), but HMM decision processes explain each others behavior |
| Angle Difference | Wildtype fed worm (N2\_f1) is most dissimilar to all other worm types |
| Angular Velocity | Similar to Angle Difference, the Wildtype fed worm (N2\_f1) is most dissimilar to all other worm types |
| Speed | Wildtype fed worm (N2\_f1) and Mutant fed worm (tph1\_f6) yielded similar results to one another |
| Speed + Angle Difference | Unfed (N2\_nf4 & N2\_nf5) and insatiable mutant (tph1\_f6) worms showed similarity, leaving the Wildtype fed (N2\_f1) worm on its own. |

**TBL10.** Summary of results

6.2 Challenges

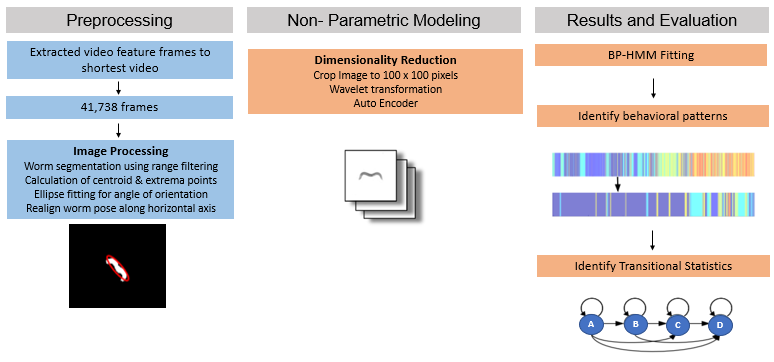
Noisy data was a challenge that we worked to overcome throughout the analysis process. Features calculated using image data, e.g. Skewer Angle, Elongation, had large fluctuations between consecutive frames, indicating errors in the segmentation of image frames. Resegmentation using alternative methods was undertaken to address this. Additionally, differences in scale, variance and stationarity made clustering difficult.

Another challenge when comparing worms is that direct path comparison is irrelevant because the starting orientation and position of the worm is not the same. Overall it is futile to compare the full trajectory of the worms because there are so many variables that could impact a worm’s decision to move into any given segment of the agar plate. Due to this we decided to work with fixed interval worm trajectories that were standardised through a centering and rotation process.

Data volume and model complexity impacted research productivity. Given runtime complexity of KML3D O(n\_tradj \* # of clusters \* rerolls \* tradj\_length) and Baum Welch O(n\_states^2 \* sequence\_length), fitting of long sequences is computationally costly. To mitigate this source frames were downsampled reducing tradj\_length with respect to KML3D and working with long interval trajectories reducing sequence\_length with respect to Baum Welch. In addition to downsampling, parallelized model training was necessary to explore hyperparameter space, and produce bagged models to overcome local minimum issues.

6.3. Future Work

Future work differentiating C. elegans using hidden Markov Modeling techniques identified during the present study will include a nonparametric approach to learning. This approach will seek to discover the presence of a more fine-grained set of hidden behavioral states and the transitional relationships between them directly from the observed data itself using Bayesian statistical inference techniques via Markov Chain Monte Carlo (MCMC) sampling. The data preprocessing for this approach, as outlined below, has already been undertaken.



**TBL11.** Summary of results

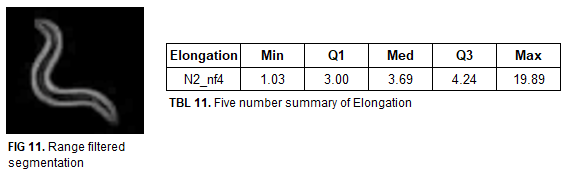
6.3.1. Data Preprocessing

Due to the modeling techniques employed for the nonparametric approach estimate the parameters of each behavioral state as a linear dynamical system, and since the number of features used to model each system affects the computational time of the iterative sampling process, it was decided that the pixel data of the image frames over time would be used as the designated features. It was also decided that the pixel data would best serve the nonparametric modeling, as the goal of this approach was to identify patterns of behavior that might not be captured entirely by modeling locomotive trajectories, but are expressed more in terms of how the worm’s body position or pose changes over time as it explores its surroundings.

6.3.1.1. Image Processing

All image processing was done using Matlab’s image processing toolbox. Representations of the pixels were generated according to the following approach: (1) image segmentation techniques were used to separate the worm from the background of the image. Because the camera on the tracker changes location as the worm moves, it creates a shifting light issue, making intensity thresholding, background subtraction, or morphological opening difficult to employ effectively. On account of this, the approach taken here was to run a sliding neighborhood operation called a range filter over the image. By considering variations in pixel intensity within a local region of the image (a statistical approach that looks at the second moment of the gray scale image, or its variance) as opposed to looking at threshold values, we were able to identify the worm based on it having more “texture (larger local variability) than the rest of the pixels in the image [6]. Range filtering serves to highlight the edges and surface contours of the worm as shown in [FIG 11]. A neighborhood range of three by three pixels was used. Holes were then filled and edges smoothed. This mask of the image was used to extract the worm from the background of the image. The worm was defined as the largest object after these procedures were performed. (2) The centroid was then identified as the center of mass of the largest object and an ellipse was fit to it to determine its angular orientation. (3) The worm was then realigned horizontally along the axis of its body using its angle of orientation and the image was the cropped to 100 x 100 pixels in size in order to make dimensionality reduction and model clustering of each state more effective.

Results from image processing were tested using N2\_nf4 worm image frames. Of the 80,785 frames our segmentation algorithm fails on only 300 or 0.4% of the time as measured by the standard Tukey method for determining outliers on the feature of Elongation, as show in [TBL 12]. below. This is a tenfold improvement on the previous algorithms used for feature generation of the datasets discussed above in Section 3. Computational complexity of texture-based image processing was linear time O(mnx), where m x n = width and height of the image and x = number of images. Runtime of the parallelized computation with a four core machine was 15 minutes for 80,785 images.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Elongation** | **Min** | **Q1** | **Med** | **Q3** | **Max** | **Outliers** | **% Dataset** |
| **Threshold** | 1.01 | 2.84 | 3.69 | 4.37 | 90.50 | 6544 | 8% |
| **Texture** | 1.03 | 3.00 | 3.69 | 4.24 | 19.89 | 300 | 0.4% |

**TBL 12.** Five Number Summary of Elongation with Outlier Results*.*

6.3.1.2. Dimensionality Reduction

To improve tractability of model computations, the next step in the approach should be to reduce the dimensions of each image. Following [5], performing a wavelet decomposition will help encode image differences at a more complex level [7]. Training an autoencoder on the transformed image data will serve to reduce the dimensions of each image without the loss of significant information that differentiates each image.

6.3.2. Model Fitting

The modeling approach chosen then takes the encoded hidden layer outputs from the trained neural network for each image as input and produces the following fitted model outputs: (1) number of shared and unique behavioral states observed between a set of different worm videos, (2) autoregressive parameters that correspond to each of the distinct behavioral states observed, (3) transition parameters that indicate how often each behavioral state comes before or after any other, and (4) for each frame in each video a classification is made for the most likely behavioral state. The particular fitting type we chose to use is a recently well established nonparametric method known as the Beta Process Hidden Markov Model. The reader is directed to [8] for formal mathematical descriptions of the model, but a brief summary of its methods and application/desirability for our data and modeling goals are as follows.

The structure of this model can be divided into two parts. (1) First, the beta process component (and the predictive distribution known as the Indian Buffet Process that is induced by marginalizing over the latent beta process) [9] is used for relating multiple time series with each other using feature vector selection. The predictive process is a culinary metaphor meant to capture the multiple class clustering phenomenon it describes. In it customers (the different time series of worm videos) arrive at an infinitely long buffet of dishes (the shared yet distinct set of behavioral states or features). How dishes are selected is described mathematically by the underlying process, where subsequent customers choose some proportion of previously tasted dishes as well as some number of new dishes. The model learns this distribution via the data such that it makes a determination as to which features in the shared set are “on” for one, or more, time series videos and which are not. (2) The Autoregressive Hidden Markov Model component (also commonly known as a Switching vector autoregressive model) enables the overall model to employ feature specific time series dynamics. Each behavioral state is modeled by its own linear dynamics or Vector Autoregressive (VAR) process and the observations evolve according to that state specific process. The set of processes a time series video can switch between is constrained by the features selected by the beta process component. The transitions between states are governed by a Hidden Markov Model.

The advantages to this approach are that any one component on their own is insufficient to model such complex time series dynamics. Traditional time series modeling is insufficient because it doesn’t capture the changing dynamics often found in complex datasets like this. The HMM is also insufficient alone as it makes a fundamental assumption that the observations it sees are conditionally independent given the latent state sequence, but often observations do have conditionally linear dynamics. The BP-HMM model also incorporates an element of bias towards self-transition, further improving the clustering of observations into behavioral states compared to other Markov switching processes. The strong advantages to the beta process component of the model for our purposes is that it encourages the sharing of dynamic behaviors so as to infer how time series relate to each other. Because of the shared nature of the parameters, it also lends to the improvement of their estimation. Due to the sparsity or constrained nature of the behavior selection, it also allows for unique behaviors to be found. Our particular datasets and domain of research involves the need for all of these things. Worms exhibit complex and uncertain dynamics, where the ground truth of latent states is unknown. Many observed movement patterns don’t appear to occur often, or are not easily discernible with the naked eye.

6.1.3. Model Output Analysis

Given that there is no ground truth knowledge with respect to the complete set of possible behavioral states and given that many behavioral states and their connected relationships to other states might not be discernible by the human eye while watching a continuous motion video recording, our proposed approach towards analyzing the output of our modeling efforts is as follows. The overarching goal will be to use the clustering of image frame sequences by the BP-HMM to look back and check corresponding images frames for those time periods, to help discern whether these states do in fact represent a coherent and observable structure of repeated behaviors. As mentioned above, the difficulty in looking back to see transitional patterns unfold in continuous time observation will be greatly aided by the ability to refer to a set of still frames aligned in succession. Since the operative assumption is that one behavioral state might be too short a duration to represent anything meaningful, but that several contiguous transitions between states do, following [5] we will seek to organize the possible relationships between the discovered behavioral states by calculating the probabilities of transition from one given state to another as a set of pairs as well as the bigram probability, or the ratio of the probability that two states occur one after the other over the total number of observed bigrams. Appropriate visualizations of such information (including a Hinton diagram and state-maps or directed probabilistic graphs) to assist in any conclusions drawn will be investigated. One of the outputs of the BP-HMM is a set of discovered hidden state classifications for each time step in the data set (in this case image frame). In order to capture the dwell time of the worm in a particular state before it transitions to another state, this output can be fed into a CTBN [16] after transforming the vector into a set of discrete random variables (one for each of the hidden states observed in the trajectory) indexed by time. While the BP-HMM produces transition probabilities between states, it does not model information about the time durations within each state before transitioning to another state or the conditional dependencies the different states have on each other, which would allow one to more accurately detect multi-state combination patterns. CTBN’s model such dependencies with respect to time directly, as well as provide a probabilistic graphical framework with which to represent such dependencies.

# **7. Acknowledgements**

We would like to acknowledge Mingfei Shao, Ian Wang and the rest of the DePaul C. elegans lab for their help throughout this process. Special thanks to Dr. Raicu and all of our fellow class members who provided excellent feedback and ideas throughout the process from methodologies to results.

# **8. References**

[1] Ferris Jabr, “The Connectome Debate: Is Mapping the Mind of a Worm Worth It?” *Scientific American*, October 2, 2012. (https://www.scientificamerican.com/article/c-elegans-connectome/)

[2] Li Huang, Hongkyun Kim, Jacob Furst, Daniela Raicu. “A Run-length Encoding Approach for Path Analysis of C. elegans Search Behavior”

[3] Gallagher T, Bjorness T, Greene R, You Y-J, Avery L, “The Geometry of Locomotive Behavioral States in C. elegans.” *PLoS ONE* 8(3): e59865. doi:10.1371/journal.pone.0059865, 2013 <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0059865>)

[4] Kang, seung-ho & Lee, Sang-Hee & Chon, Tae-Soo, “Exploring the Behavior of Caenorhabditis Elegans by Using a Self-organizing Map and Hidden Markov Model.” *Journal of the Korean Physical Society*. 60. 604-612. 10.3938/jkps.60.604. 2012 <https://www.researchgate.net/publication/237012153_Exploring_the_Behavior_of_Caenorhabditis_Elegans_by_Using_a_Self-organizing_Map_and_Hidden_Markov_Model>)

[5] Alexander B. Wiltschko, Matthew J.Johnson, Giuliano Iurilli, Ralph E.Peterson, Jesse M.Katon, Stan L.Pashkovski, Victoria E.Abraira, Ryan P.Adams, Sandeep Robert, “Mapping Sub-Second Structure in Mouse Behavior”, *Neuron*, Volume 88, Issue 6, 16 December 2015, Pages 1121-1135

[6] Gonzalez, R.C., R.E. Woods, S.L. Eddins, Digital Image Processing, New Jersey, Prentice Hall, 2003, Chapter 11.

[7] Mallat, S.G., A theory for multiresolution signal decomposition: the wavelet  
representation. Pattern Analysis and Machine Intelligence 1989, IEEE Transactions on 11, 674-693.

[8] Emily B. Fox, Michael C. Hughes, Erik B. Sudderth, Michael I. Jordan, Joint modeling of multiple time series via the beta process with application to motion capture segmentation, Annals of Applied Statistics 2014, Vol. 8, No. 3, 1281-1313.

[9] Romain Thibaux, Michael I. Jordan, Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, PMLR 2:564-571, 2007.

[10] Mingfei Shao, Yiyang Wang, Jennifer Paine, Xiao Lin, Hongkyun Kim, Jacon First, Daniella Raicu, “*Computational Annotation of C. elegans Movement Behavior.”*

[11]K. Moy, W. Li, H. P. Tran et al., “*Computational methods for tracking, quantitative assessment, and visualization of C. elegans locomotory behavior*,” PLoS ONE, vol. 10, no. 12, Article ID e0145870, 2015.

[12] Genolini, Christophe, Xavier Alacoque, Mariane Sentenac, & Catherine Arnaud. "*kml and kml3d: R Packages to Cluster Longitudinal Data*." *Journal of Statistical Software* [Online], 65.4 (2015): 1 - 34. Web. 17 Feb. 2018

[13] Junji Yamato, Jun Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden Markov model, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, July 1992, J76-D-II:379-385.

[14] Emily B. Fox, Bayesian Nonparametric Learning of Complex Dynamical Phenomena, Doctoral Thesis, Massachusetts Institute of Technology, 2009.

[15] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, & Alan S. Willsky, An HDP-HMM for Systems with State Persistence, International Conference on Machine Learning (ICML), 2008.

[16] Uri Nodelman, Christian R. Shelton, Daphne Koller, Continuous Time Bayesian Networks, Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, p.378-387, August 01-04, 2002.

[17] T. Calinski and J. Harabatz, “A dendrite method for cluster analysis,” Comm. in Statistics, vol. 3, no. 1, pp. 1–27, 1974.