Generation of Simulated Data and Model Accuracy

Josh Cullen

17 January 2020

Background

As part of the upcoming manuscript that compares our newly developed method against frequently used methods, we will be testing them using both simulated and empirical data. These tests on simulated tracks will be performed to identify and recover pre-specified activity centers (ACs) and pre-specified behavioral states. There are a wide variety of options available in terms of simulating animal movement, whether based on empirical distributions or by using parametric distributions. In other cases, simulations can be generated from a process model that already has built in constraints (hidden Markov models via the *momentuHMM* package). However, the available packages for generating *de novo* simulations (*adehabitatLT, trajr, waddle*) do not appear suitable or flexible enough for what we want to do. Therefore, I developed my own code to generate a multi-state correlated random walk (CRW) model of three different behaviors ('resting', 'exploratory', 'transit') as well as a multi-phase biased random walk (BRW) that could be used to simulate biased movement towards activity centers (ACs). These models were based upon established CRW and BRW models, but were modified to achieve the desired result.

Simulations

BRW for identifying ACs

First, 10 ACs will be randomly identified for which to build the simulated BRW trajectory. Next, values will be chosen for the number of observations per time segment (n), the number and location of ACs (Z.centers), the number of phases or time segments (nphases), the initial location (Z0), the shape (a) and scale (b) parameters for a gamma distribution from which to draw step lengths, and the concentration parameter (ρ) from which to draw the turning angles:

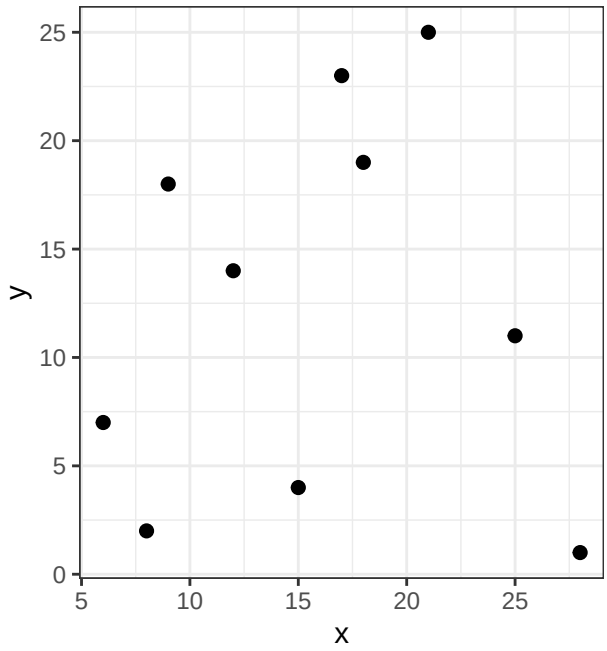

Figure 1: Location of the 20 randomly selected ACs.

```
set.seed(3)
track<- multiBRW.sim(n=1000, a = 1, b = 1, nphases = 15, Z.center = AC, Z0 = c(3,4), rho = 0.8)
track$time1<- 1:nrow(track)  #add variable for time
```

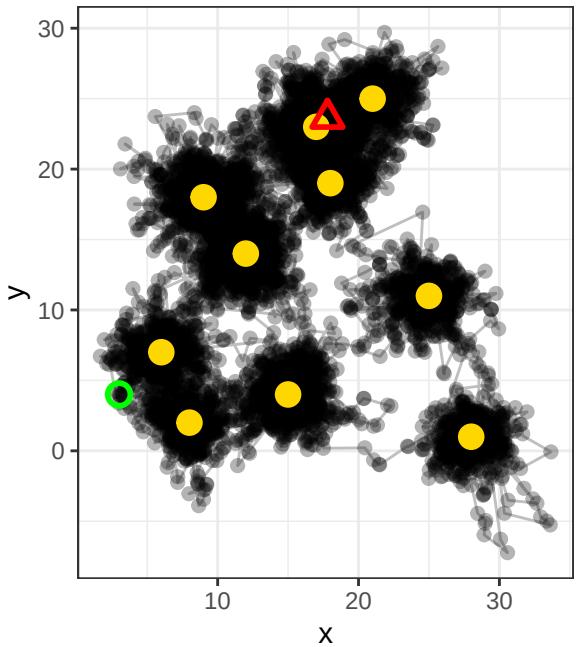

Figure 2: Location of the 10 randomly selected ACs (gold) with respect to the trajectory simulated from a BRW (grey). The initial location is denoted by the green circle and the ending location by the red triangle.

These observations from the simulated trajectory will be used to create a grid on which to discretize space before running the segmentation algorithm. Cell resolution will be 5 units.

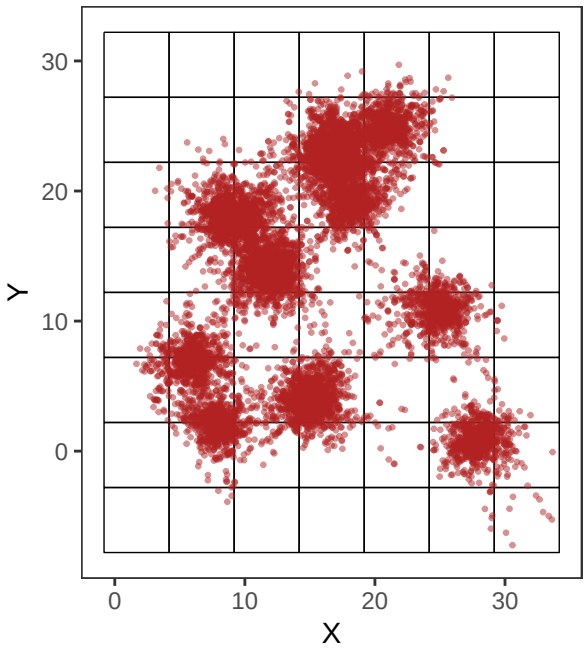

Figure 3: Point locations from simulated trajectory over discrete space.

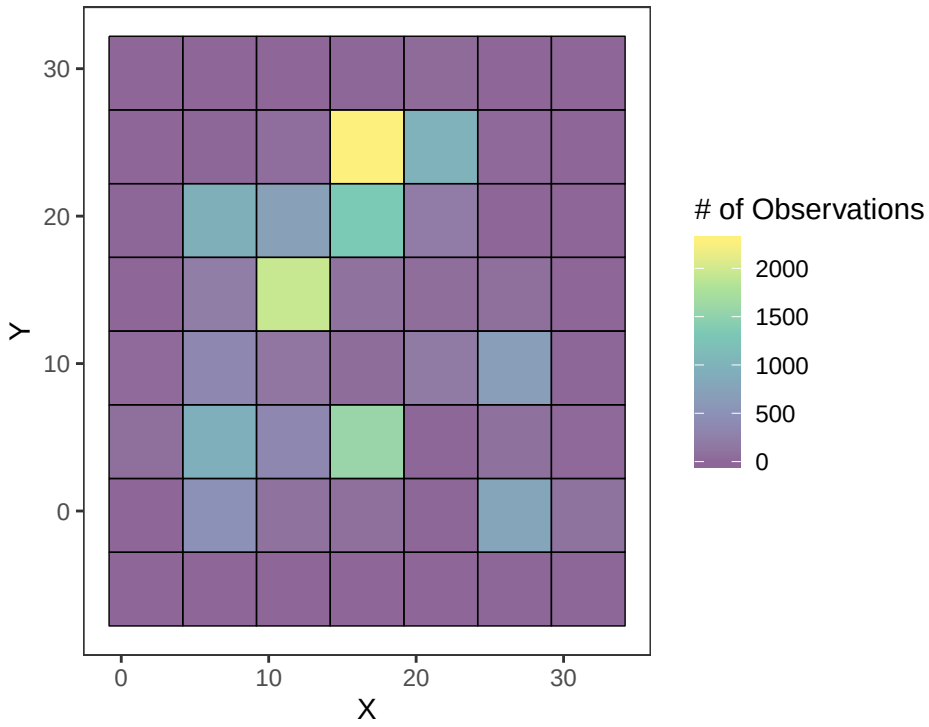

Figure 4: Density surface of point distribution.

Now with all observations assigned to a particular grid cell, the segmentation algorithm is used to identify breakpoints within the time series of movement:

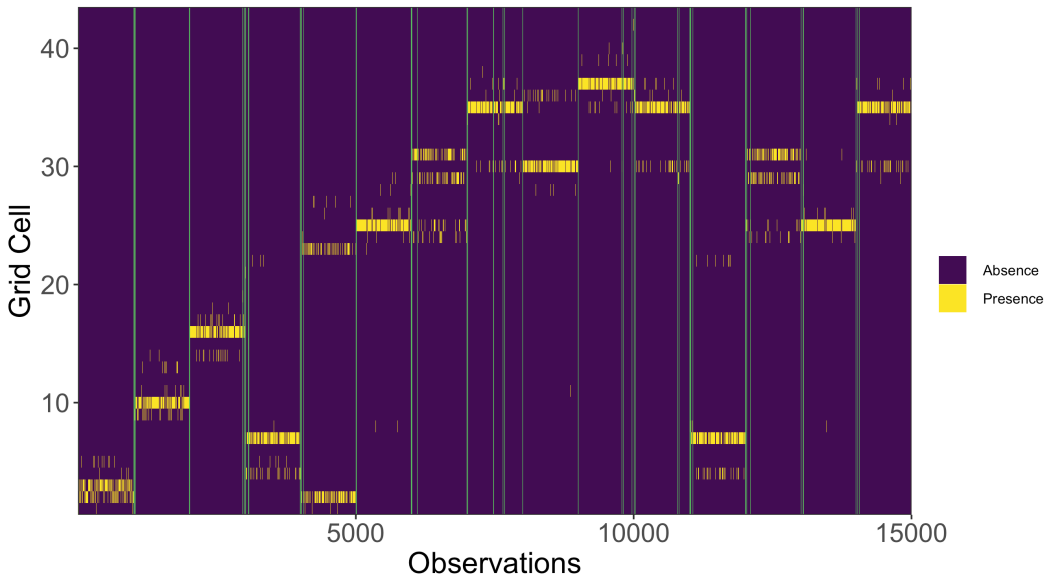

Figure 5: Heatmap of grid cell occupancy over 15000 observations. Vertical green lines denote modeled breakpoints.

There are many breakpoints identified by the segmentation model, almost four-fold greater than the true number of breakpoints. When compared against one another, most of the breakpoints from the model overlap with true breakpoints.

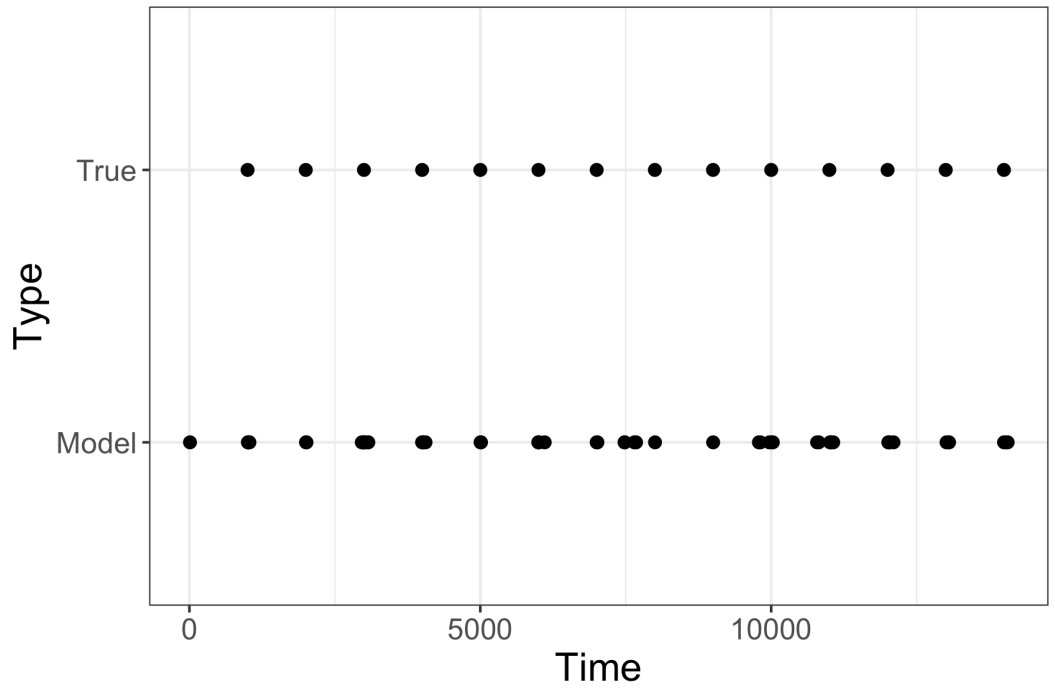

Figure 6: Comparison of true breakpoints from the BRW simulation compared to those identified by the model. N = 14 for true breakpoints and N = 53 for modeled breakpoints.

Although there are some modeled breakpoints that occur within the middle of a time segment, most fall on or next to the true breaks in the time segments. Next, the raw observations will be assigned to these segments, which will serve as input for the mixture model used to identify and assign ACs.

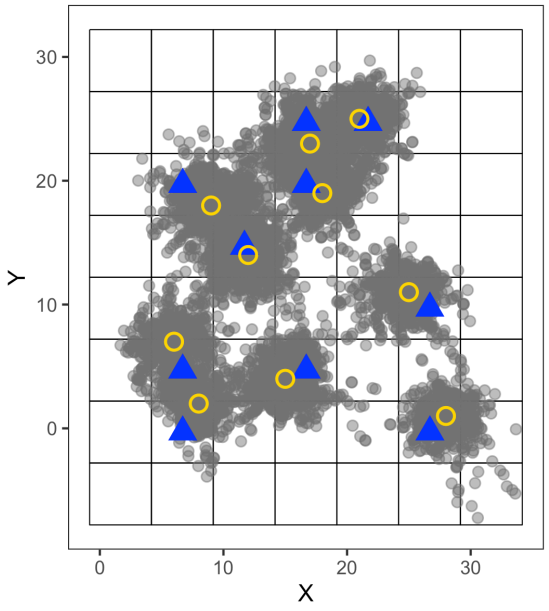

Figure 7: Comparison of true ACs (gold) to modeled ACs (blue) over the raw observations (grey).

From these results, it appears that the model did a good job of approximating the correct location of the ACs since it was limited to the centroids of occupied grid cells. Now that the ACs have been correctly identified, the next step is to determine the accuracy of AC assignment to the observations from all of the time segments. In our example, the model is 98.7% accurate in the assignment of ACs to each of the 15000 observations. The accuracy of AC locations and assignments to observations is dependent on the grid size and resolution, necessitating the testing of multiple grid sizes to find the best fit to the data.

CRW for identifying behaviors

Since our model uses a mixed-membership model to identify the proportion of behaviors within each time segment, I simulated tracks using both hard-clustering and mixed-membership methods to test the model's ability to classify these different types of data. Both simulations required the user to define the length of each time segment (n), provide a vector of the behaviors (behav), a vector of the shape and scale parameters used to generate step lengths (SL.params), a vector of the mean angles and concentration parameters used to generate the turning angles (TA.params), and the initial location (Z0). A gamma distribution was used to generate step lengths while a wrapped Cauchy distribution was used to generate turning angles. For the hard-clustering method, the 'behav' vector is of length equal to the number of time segments where there is a greater probability of being in either a 'resting' or 'exploratory' behavior than in a 'transit' behavior. In the mixed membership model, this first step is the same, but another one is performed within each time segment where the original behavior assigned to each segment has the greatest probability of occurrence compared to the other two.

Hard-clustering

First, I will simulate data using the hard-clustering method for behavior and subsequently run it on the model to evaluate accuracy of behavior classification.

```
#define behaviors and randomly sample 50 (for 50 time segments)
#weight probs so that behavior 1 (Resting) occurs 50%, behavior 2 (Exploratory) occurs 35%,
#and behavior 3 (Transit) occurs 15%
set.seed(1)
behav<- sample(c(1,2,3), 50, replace = TRUE, prob = c(0.5, 0.35, 0.15))
n=50
SL.params<- data.frame(shape=c(0.25, 2, 10), scale = c(1, 1, 1))
TA.params<- data.frame(mu=c(pi, pi, 0), rho = c(0.8, 0, 0.8))

track<- CRW.sim(n=n, behav = behav, SL.params = SL.params, TA.params = TA.params, Z0=c(0,0))
```

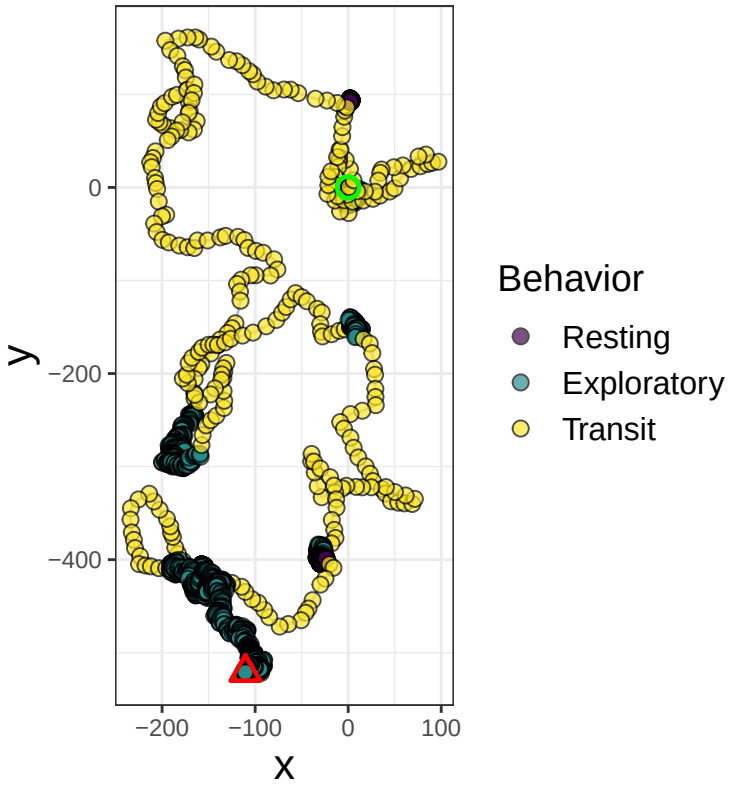

Figure 8: Simulated CRW track with a single behavior assigned for each time segment. The green circle indicates the starting location and the red triangle is the ending location.

We can also compare the SL and TA distributions among behaviors:

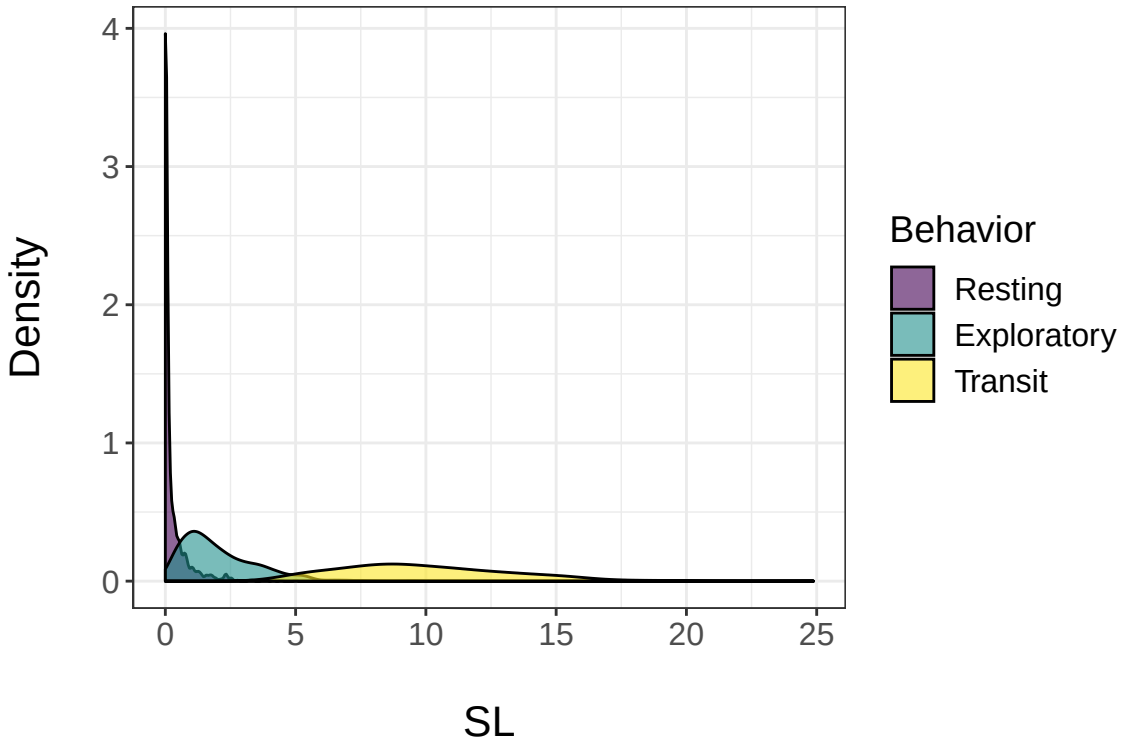

Figure 9: Distributions of step lengths.

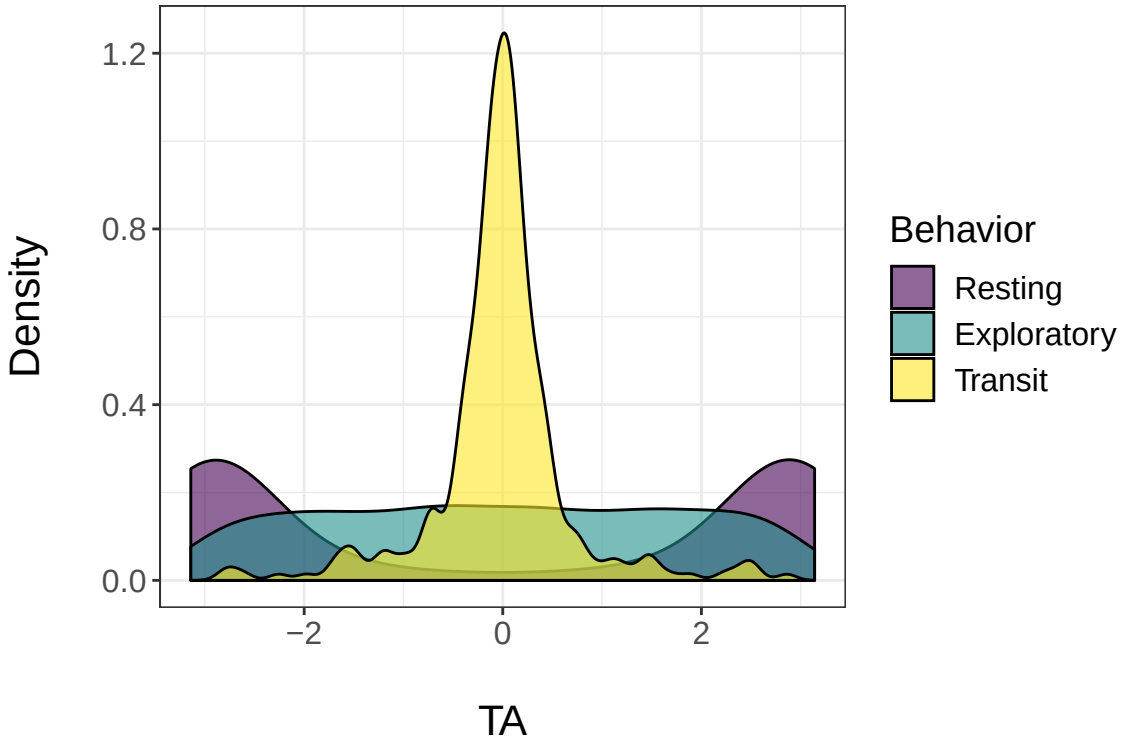

Figure 10: Distributions of turning angles.

From this simulated track, we can run the time segmentation algorithm to discern the number and locations

of the breakpoints in the time series of SL and TA. These breakpoints are lined up with the data, as well as compared against the true breakpoints:

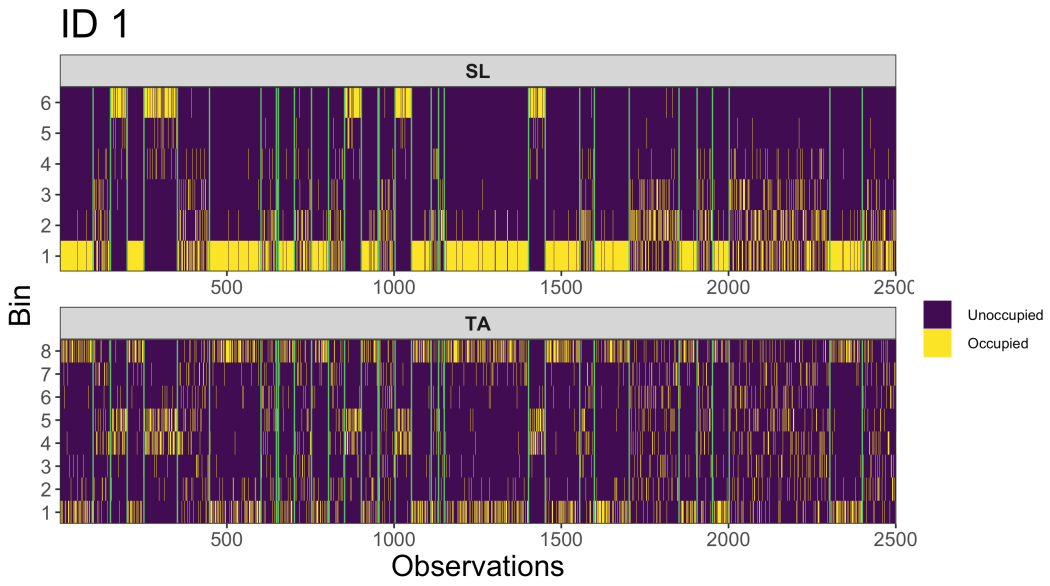

Figure 11: Heatmap showing step lengths and turning angles for each observation over time as characterized by previous binning decisions. Vertical green lines indicate breakpoints determined by the model.

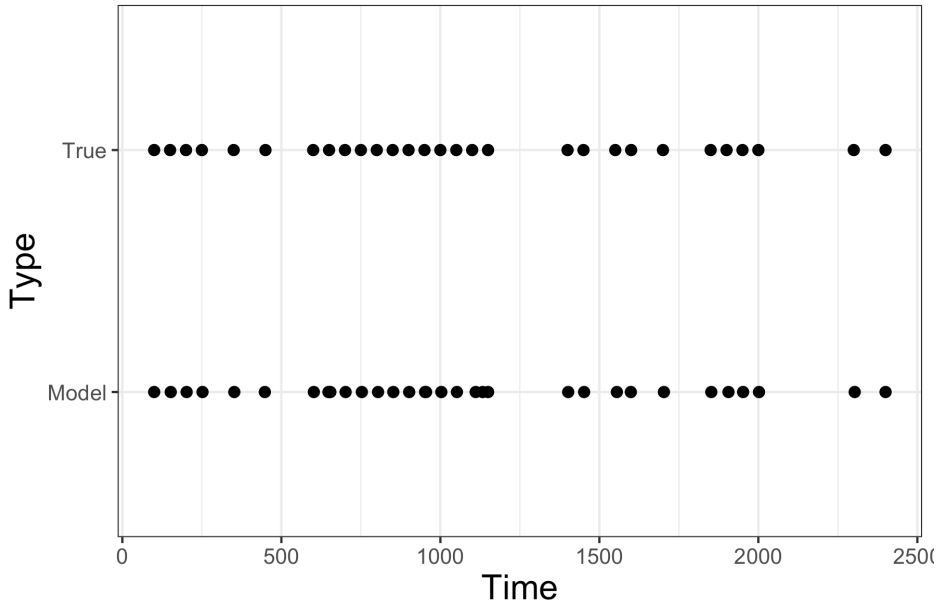

Figure 12: The model identifies a few more breakpoints than the true number of breakpoints. The locations of the modeled breakpoints are very similar if not identical to the true positions.

At a glance, the segmentation appears to do a good job compared to the true breakpoints. Next, I will run the latent Dirichlet allocation (LDA) model to cluster the time segments into separate behaviors. Following

this analysis, I will determine the likely number of behavioral states from the posterior distribution, assign behavioral states to each cluster, and plot these over time and geographic space.

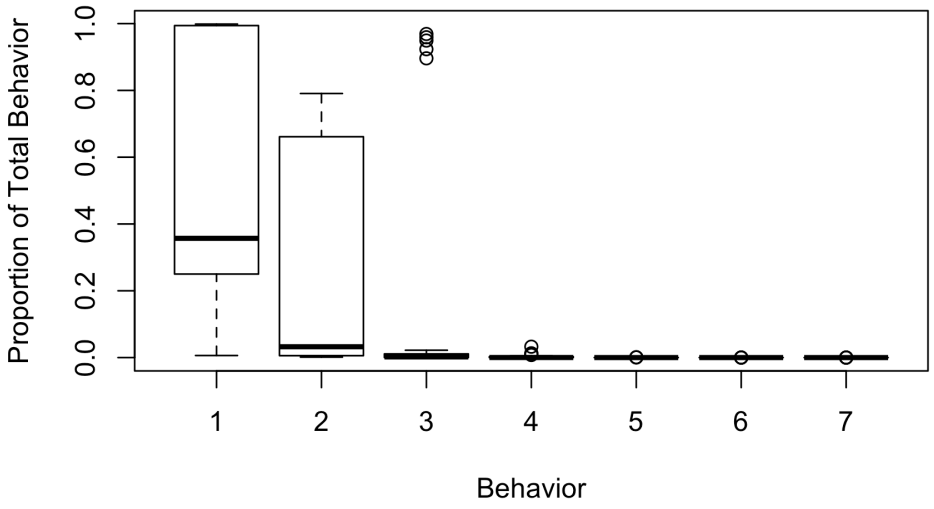

Figure 13: Probability of behavioral assignment for all time segments.

It looks like there are likely only three behaviors being identified, but we can look at the numbers and compare this to the true proportions of the simulated data.

Model results:

```
## [1] 0.538 0.314 0.146 0.003 0.000 0.000 0.000
```

True proportions of behaviors:

```
## behav
##    1    2    3
## 0.46 0.42 0.12
```

The modeled proportions appear to be slightly skewed compared to the true values used to generate the simulation. Next, let's look at the histograms for each of these top three clusters/behaviors from the model and assign a state to each:

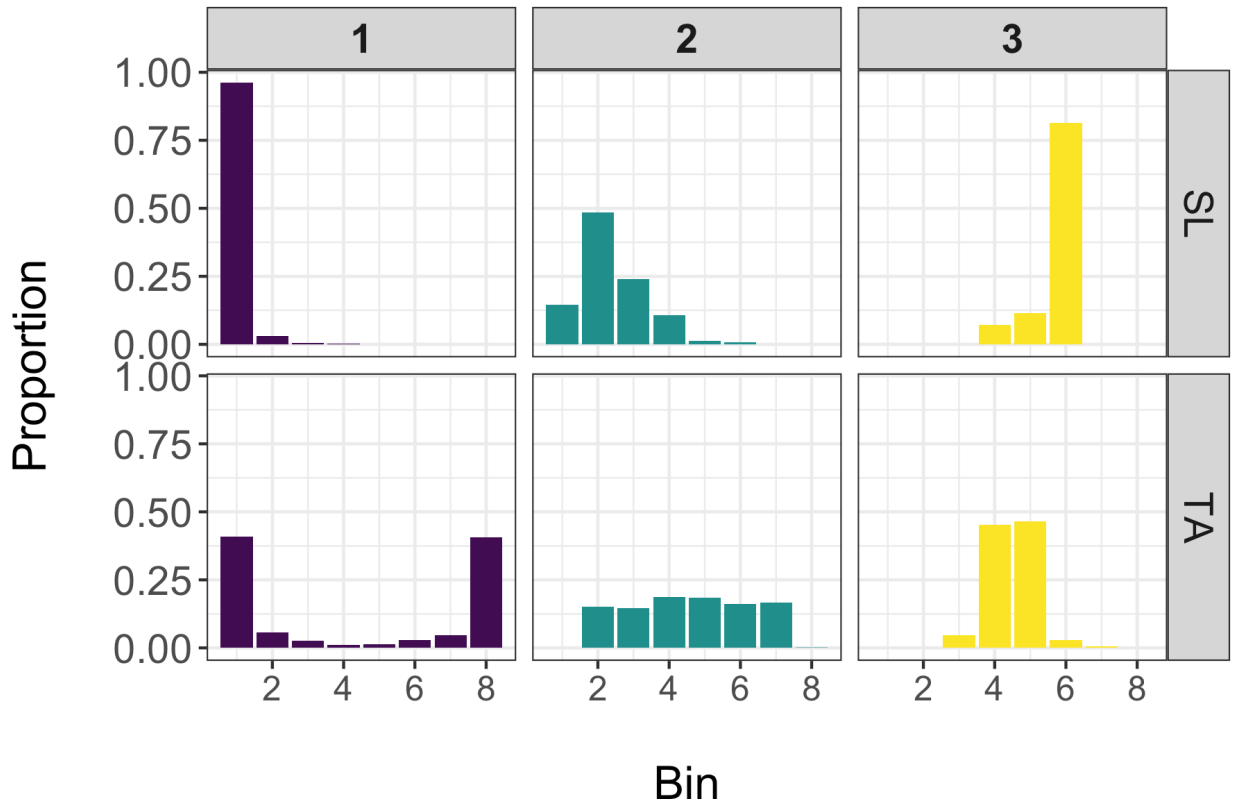

Figure 14: Distributions for the SL and TA of each of the top three clusters from the LDA model.

These histograms are markedly different in both their SL and TA and appear to show 'resting', 'exploratory', and 'transit' behaviors, respectively. Now, we can investigate how these behavioral estimates match with the true behaviors from the hard-clustering simulation:

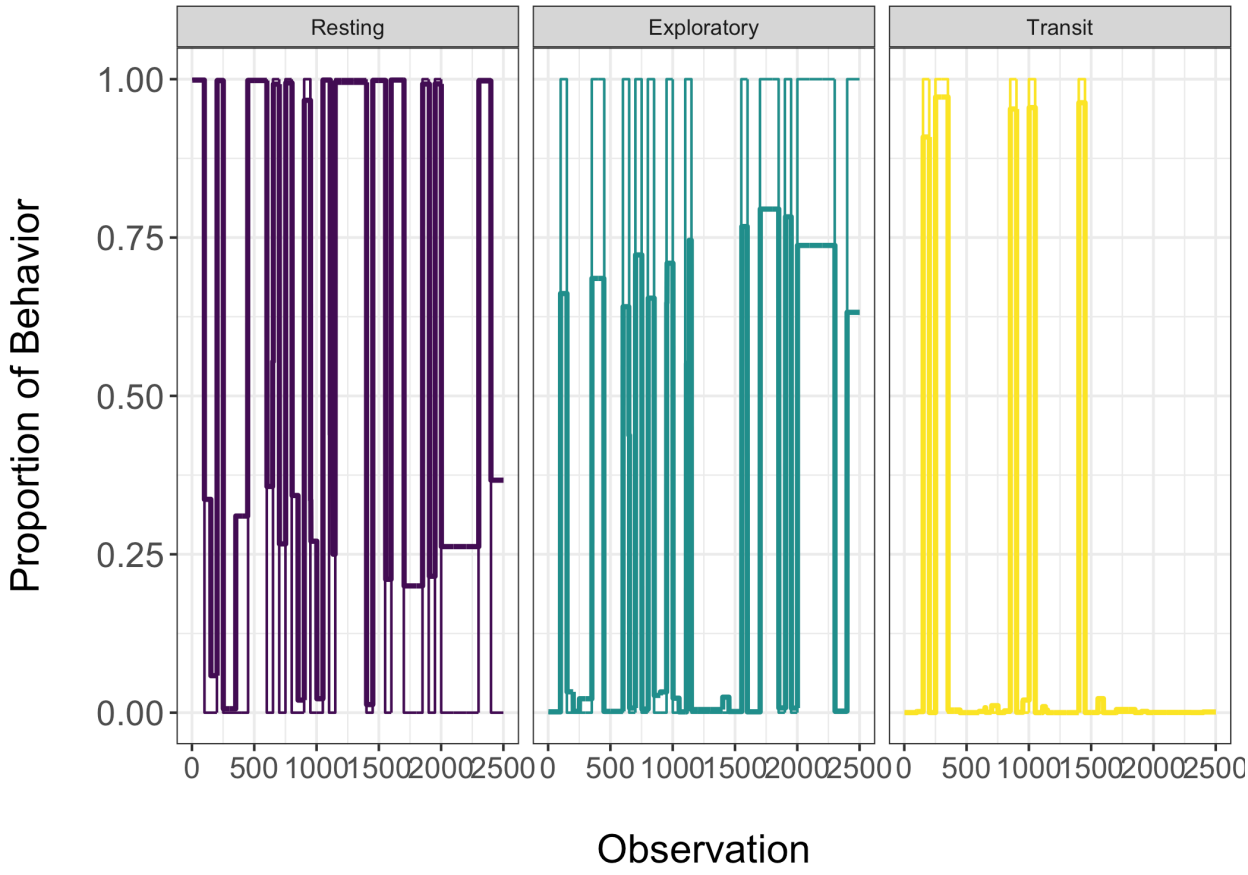

Figure 15: Each of these panels shows a separate behavior, where the proportion of the true state occurrence is indicated by the thinner line and the modeled proportion is denoted by the thicker line.

Results for all three behaviors show that the model does a pretty good job at estimating the proportion of each behavior across all time segments. However, it appears that the 'resting' and 'transit' behaviors were assigned properly more often compared to the 'exploratory' behavior. This could be a result of these behaviors being relatively similar in their SL and TA.

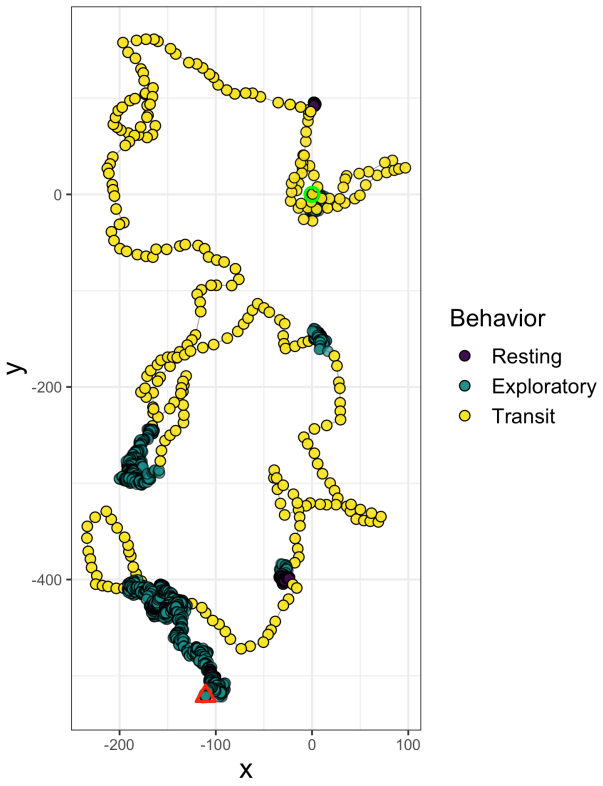

Figure 16: Simulated hard-clustering CRW track with behaviors estimated by the model. The green circle indicates the starting location and the red triangle is the ending location.

Outwardly, the behavioral states on this track as estimated by the model look almost identical to the original simulation. To quantify the model's accuracy, we will directly compare the behavioral estimates for all time segments of the track.

The overall accuracy of the model was 97.7% (including all behaviors together). When breaking this down by behavior, the 'exploratory' behavior was identified with the least accuracy (96.4% accurate), which was still quite high. The 'transit' behavior was the next most accurate (98.0%) followed by the 'resting' behavior (98.9%). It appears that the 'resting' and 'transit' behaviors were easily identified since they were much different from each other and represented extremes. The 'exploratory' behavior had a somewhat similar SL distribution compared to the 'resting' behavior, but had a uniform TA distribution. This resulted in the proportion of 'exploratory' behavior assigned to specific time segments to be lower than thier true value. However, the accuracy of this model is very high when evaluating all behaviors together or separately.

Mixed-membership clustering

Now, I will simulate data using the mixed-membership clustering method for behavior and subsequently run it the same as was done for the hard-clustering simulation.

```
#Define behaviors and randomly sample 50 (for 50 time segments)
#Weight probs so that behavior 1 (Resting) occurs 50%, behavior 2 (Exploratory) occurs 35%,
#and behavior 3 (Transit) occurs 15%
```

```r
#Within each behavior segment, dominant behavior prob is 0.8 while other two are 0.1

set.seed(2)

#create vector of dominant behaviors per time segment (50 segments)
behav<- sample(c(1,2,3), 50, replace = TRUE, prob = c(0.5, 0.35, 0.15))

#randomly choose 3 segments of each behavior to be 'pure' instead of mixed
behav1.pure<- sample(which(behav==1), 3, replace = FALSE)
behav2.pure<- sample(which(behav==2), 3, replace = FALSE)
behav3.pure<- sample(which(behav==3), 3, replace = FALSE)

pure<- c(behav1.pure, behav2.pure, behav3.pure) %>% sort()


#create vector of behaviors within each time segment (duration of 100 steps)
behav.full<- vector("list", length(behav))
for (i in 1:length(behav)) {
  if (i %in% pure) {
    behav.full[[i]]<- rep(behav[i], 100)
  }else if (behav[i] == 1) {
    behav.full[[i]]<- sample(c(1,2,3), 100, replace = TRUE, prob = c(0.8, 0.1, 0.1))
  } else if (behav[i] == 2) {
    behav.full[[i]]<- sample(c(1,2,3), 100, replace = TRUE, prob = c(0.1, 0.8, 0.1))
  } else if (behav[i] == 3) {
    behav.full[[i]]<- sample(c(1,2,3), 100, replace = TRUE, prob = c(0.1, 0.1, 0.8))
  }
}
behav.full<- unlist(behav.full)

SL.params<- data.frame(shape=c(0.25, 2, 10), scale = c(1, 1, 1))
TA.params<- data.frame(mu=c(pi, pi, 0), rho = c(0.8, 0, 0.8))

track<- CRW.sim(n=1, behav = behav.full, SL.params = SL.params, TA.params = TA.params, Z0=c(0,0))
```

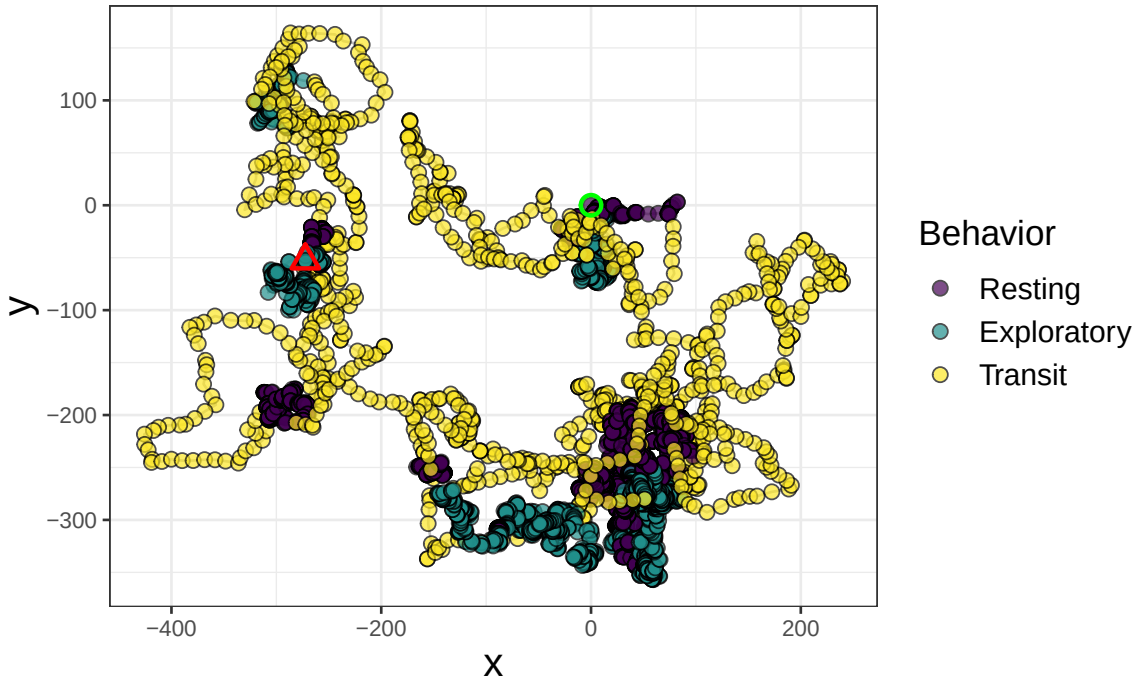

Figure 17: Simulated mixed-membership CRW track showing single dominant behavior assigned for each time segment. The green circle indicates the starting location and the red triangle is the ending location.

We can also compare the SL and TA distributions among behaviors at both a fine (by observation) and coarse scale (by time segment):

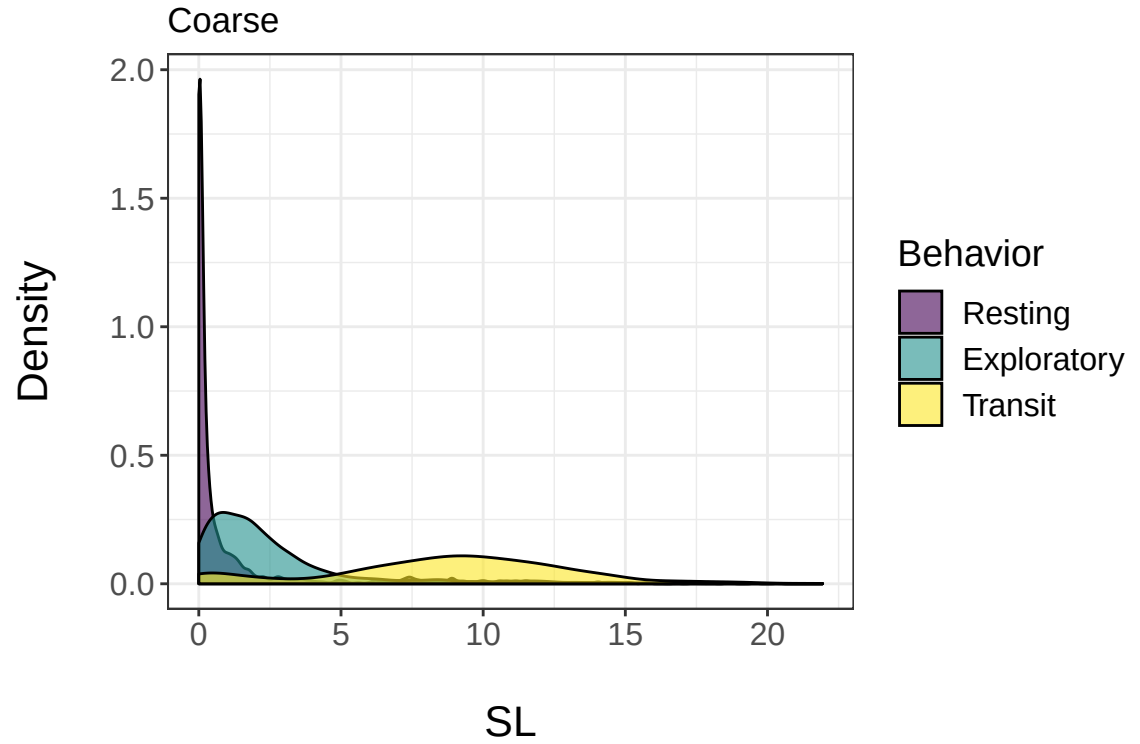

Figure 18: Distributions of step lengths at coarse scale.

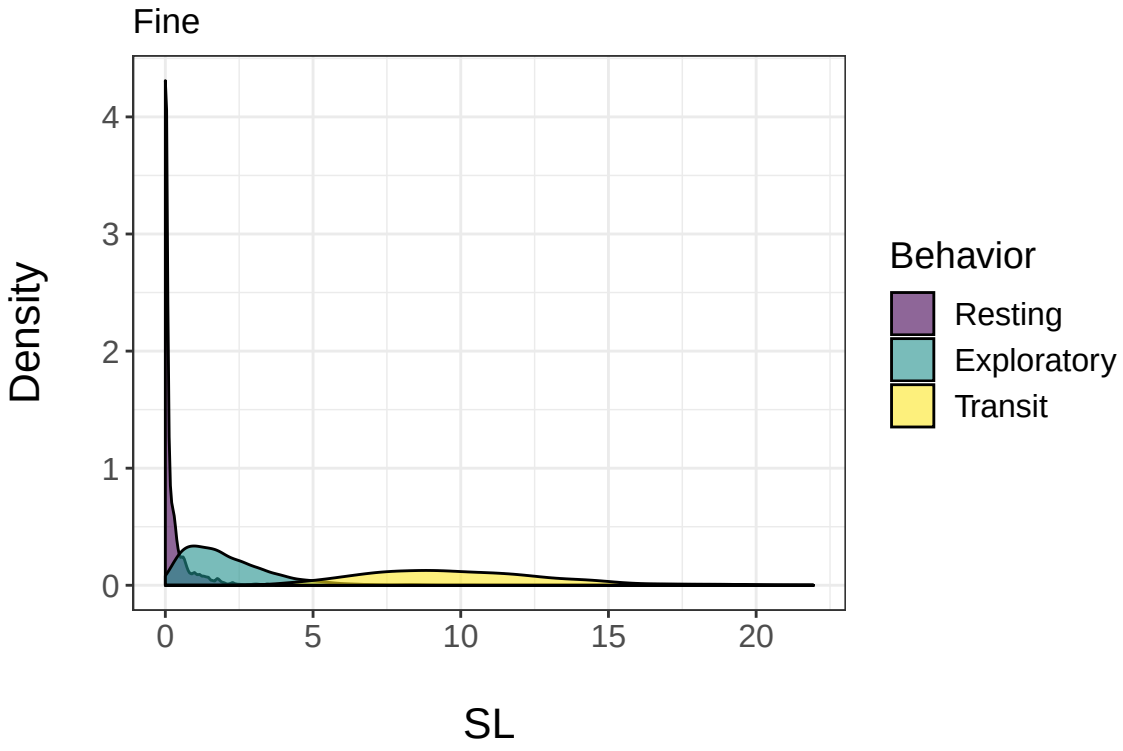

Figure 19: Distributions of step lengths at fine scale.

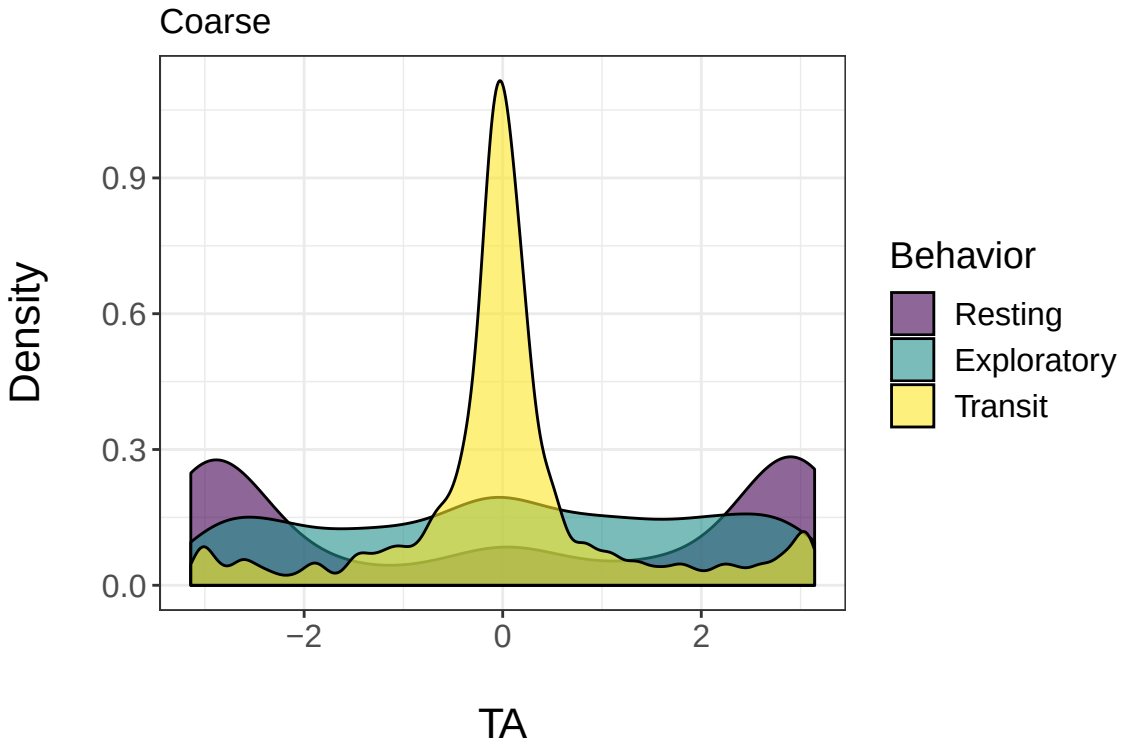

Figure 20: Distributions of turning angles at coarse scale.

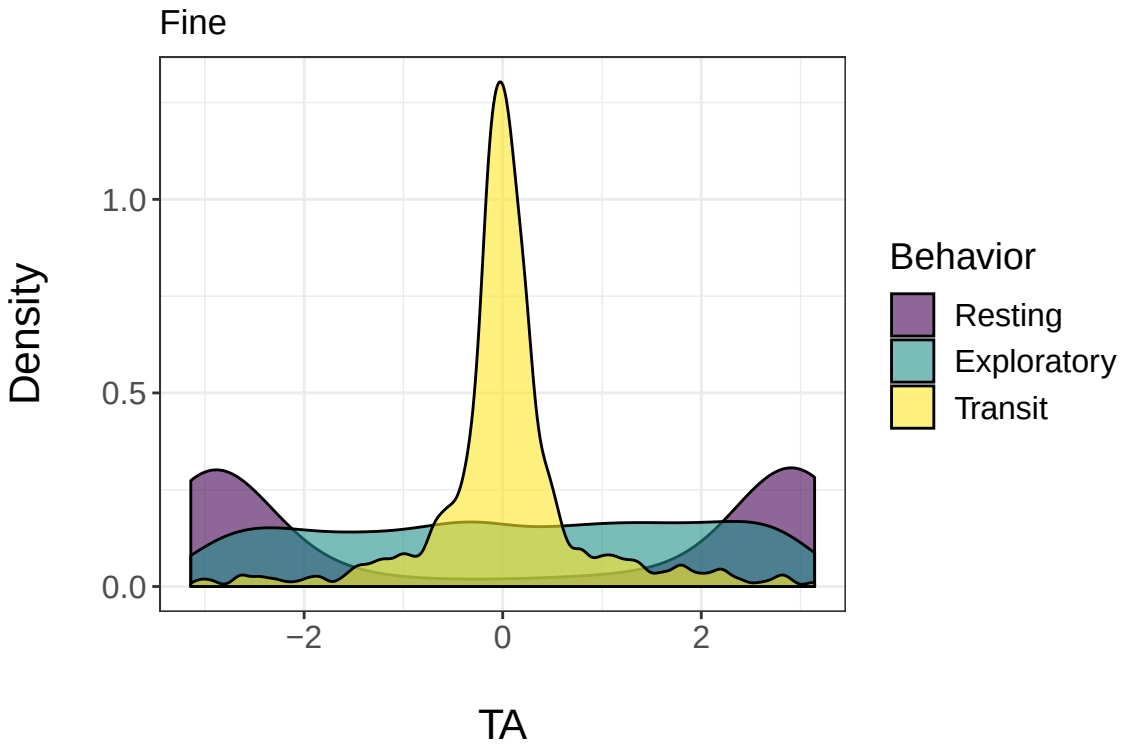

Figure 21: Distributions of turning angles at fine scale.

Again, we can run the segmentation algorithm to determine the number and locations of the breakpoints in

the time series of SL and TA. These breakpoints are also compared against the true breakpoints:

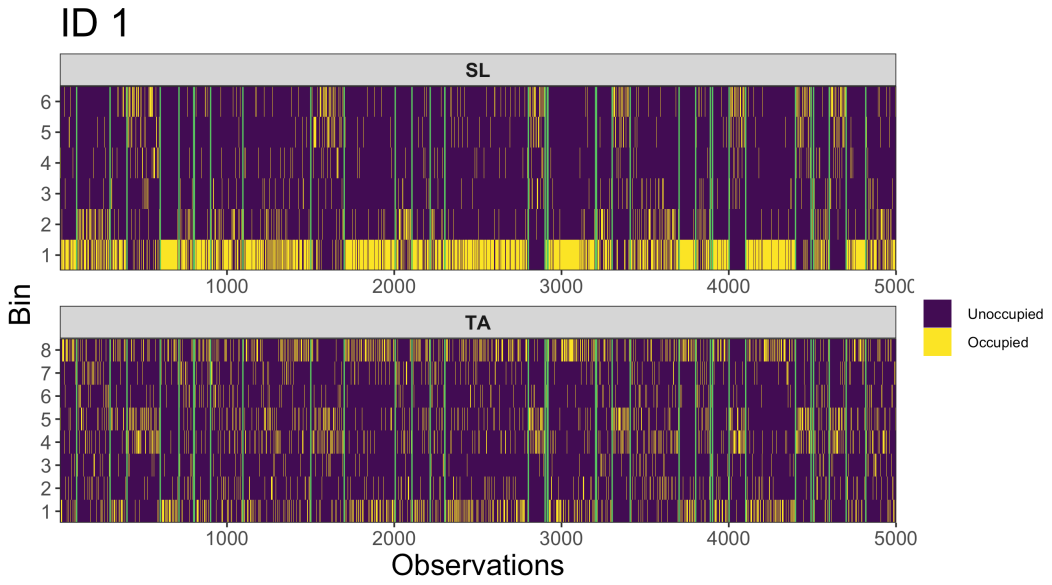

Figure 22: Heatmap showing step lengths and turning angles for each observation over time as characterized by previous binning decisions. Vertical green lines indicate breakpoints determined by the model.

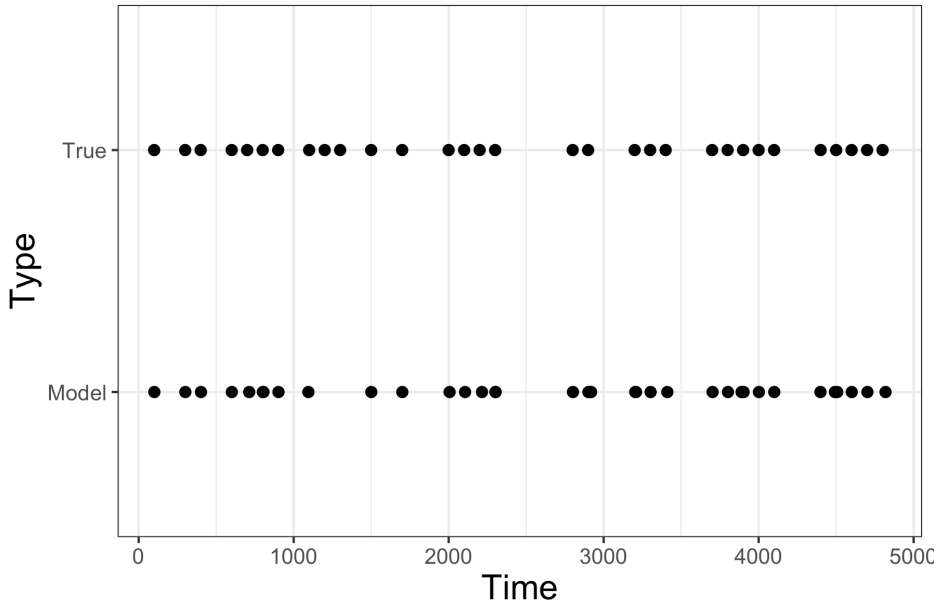

Figure 23: The model identifies slightly more breakpoints than the number of true breakpoints. The locations of the modeled breakpoints are very similar if not identical to the true positions.

At a glance, the segmentation method appears to do a good job compared to the true breakpoints, only missing a couple breakpoints between the 1000th and 1500th observations. Next, I will run the latent Dirichlet allocation (LDA) model to cluster the time segments into separate behaviors.

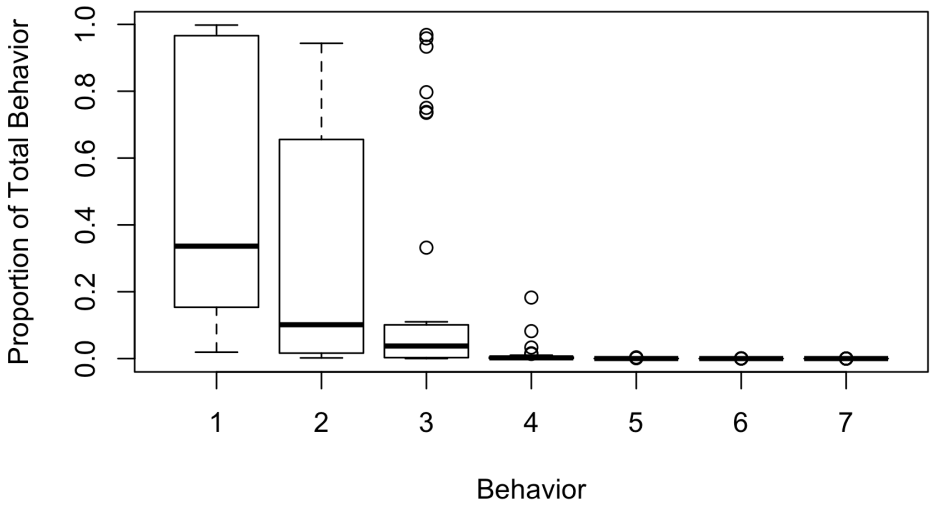

Figure 24: Probability of behavioral assignment for all time segments.

It looks like there are likely only three behaviors being identified, matching the number of true behaviors that we simulated. We can look at the numbers and compare this to the true proportions of the simulated data.

Model results:

```
## [1] 0.502 0.293 0.194 0.011 0.000 0.000 0.000
```

True proportions of behaviors:

```
## behav
##    1    2    3
## 0.50 0.32 0.18
```

The modeled proportions appear to be very similar to the true proportions if they are assigned in the same order by the model. Next, let's look at the histograms for each of these top three clusters/behaviors from the model and assign a state to each:

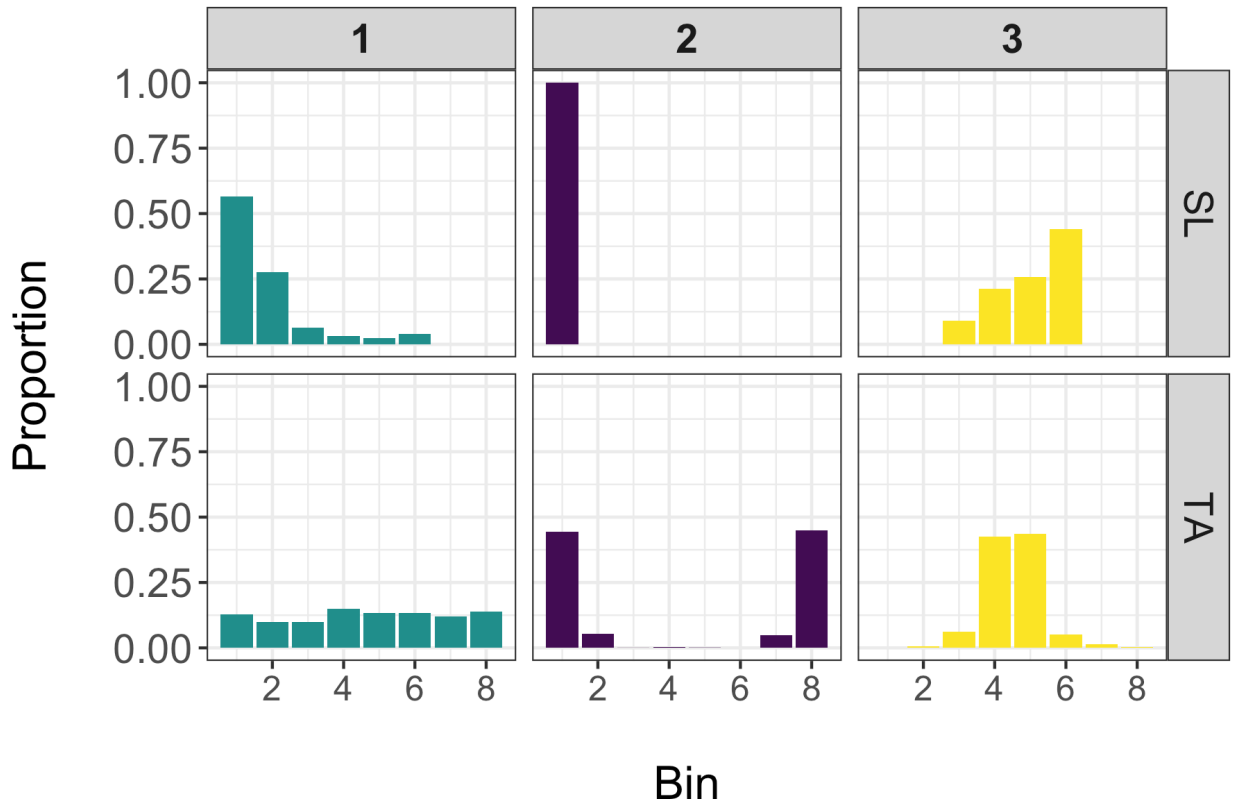

Figure 25: Distributions for the SL and TA of each of the top three clusters from the LDA model.

These histograms appear to be quite distinct from one another and match up with the original distributions of the mixed-membership simulation. In this figure, the 'exploratory' behavior is the first cluster, 'resting' is the second cluster, and 'transit' is the third cluster. Now, we can investigate how these behavioral estimates match with the true behaviors from the mixed-membership simulation:

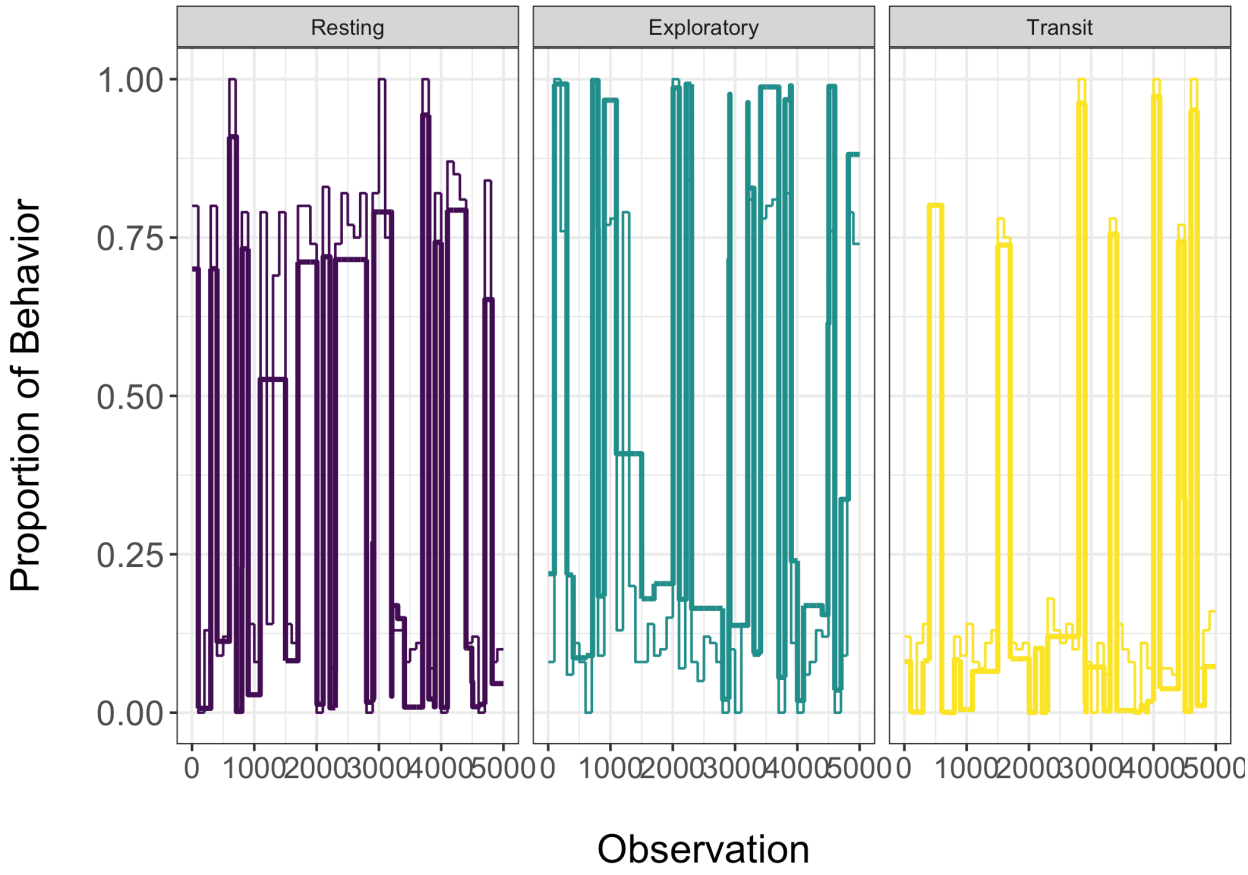

Figure 26: Each of these panels shows a separate behavior, where the proportion of the true state occurrence is indicated by the thinner line and the modeled proportion is denoted by the thicker line.

Results for all three behaviors show that the model performs very well at estimating the proportion of each behavior across all time segments. The estimated and true proportions for behaviors in each time segment are similar in most cases, demonstrating the power of the LDA to determine accurate proportions. These results may vary depending on the number and similarity of behaviors given the parameters on which behavior is originally segmented.

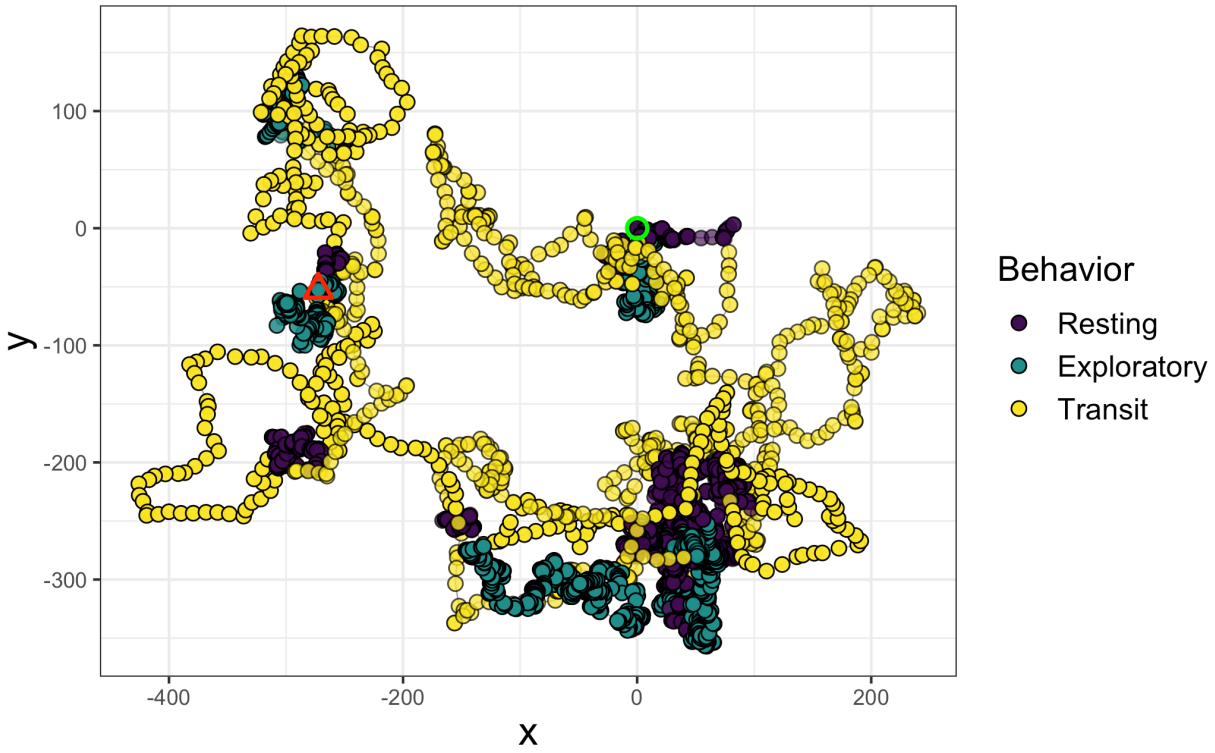

Figure 27: Simulated mixed-membership CRW track with behaviors estimated by the model. The green circle indicates the starting location and the red triangle is the ending location.

Outwardly, the behavioral states of this track as estimated by the model are very similar to the original simulation. To quantify the model's accuracy, we will directly compare the behavioral estimates for all time segments of the track.

The overall accuracy of the model in identifying the dominant behavior per time segment was 95.8% (including all behaviors together). When breaking this down by behavior, the 'exploratory' behavior was identified with the least accuracy (89.7% accurate), which was much lower than for the hard-clustering simulation (96.4%). The 'resting' and 'transit' behaviors were both more accurate (98.7 and 98.4%, respectively) than the 'exploratory' behavior, which was nearly identical to the classification accuracy of both behaviors from the hard-clustering simulation. It appears that the 'resting' and 'transit' behaviors were more easily identified since they were much different from each other and represented extremes. As an intermediate behavior type that resembled the 'resting' behavior in SL after the segmentation step of the model, the 'exploratory' behavior was not accurately classified as often. Overall, this model performed very well at behavior classification when the data were generated from a hard-clustering or mixed-membership CRW simulation.