# THE ECOLOGY OF MOVEMENT AND BEHAVIOUR: A SATURATED TRIPARTITE NETWORK FOR DESCRIBING ANIMAL CONTACTS

Supplementary Materials

Kezia Manlove, Christina Aiello, Pratha Sah, Bree Cummins, Peter J. Hudson, Paul C. Cross

August, 2018

## Contents

# 1 Quantifying information sharing in $(I, S, T)$

In this section, we provide more detail on the mutual information approach taken in *The tripartite model links animal behaviour to network projections* of the main text.

## 1.1 Special considerations due to the structure of $(I, S, T)$

The full tripartite model can be constructed from a dataframe in which rows are individual location events, with columns indicating identity of the individual involved, the coordinates of the individual's location, and the date and time the record was collected. This data structure has a few special features that merit specific mention. We imagine a dataset with $N_{\text{Observations}}$ total observations (so, $N_{\text{Observations}}$ rows). First, the data are of mixed-type, since individual identity ($I$) is a discrete, categorical (but not ordinal) variable denoting $N_{\text{Individuals}}$ different individual identities. In some cases, it may be convenient to expand $I$ to an $N_{\text{Observations}} \times (N_{\text{Individuals}} - 1)$ matrix of indicator variables. Location in space ($S$) is nearly always denoted as a multidimensional vector (e.g., $(X, Y)$ where $X$ and $Y$ denote longitude and latitude, respectively). Time ($T$) can be fully represented in one dimension. The higher-dimensional representations of $I$ and $S$ could be bound to $T$ to generate a fully expanded $(I, S, T)$ matrix, which we refer to below as $\mathbf{M}$. We note that the information in this multidimensional representation needs to rescaled so that all columns in the indicator individual set are compared on an equal footing to two- or three-dimensional description of $S$, and the one-dimensional $T$.

## 1.2 Why take an information theoretic approach?

As described in the main text, one utility of the tripartite model is its ability to capture how behavioral motifs constrain correlation structures within (and thus the dimension of) $(I, S, T)$. Practically speaking, the correlation structure can be captured by quantifying the information shared among the $I$,$S$, and $T$ variables. We could quantify shared information in a variety of ways, including through classic (albeit linear) dimension reduction methods like Principal Coordinates Analysis (PCoA) [1], or through an ensemble of generalized linear models (GLMs) in which each column is modeled in response to the others. However, the PCoA approach cannot readily capture the $S \times T$ relationship that characterizes group size, and the GLM-based approaches are problematic because they do not allow direct comparison among different metric combinations. For example, the Akaike Information Criterion for a model fitting $I$ as a function of $S$ and $T$ is based on a fundamentally different data structure than a model that treats $S$ as a function of $I$ and $T$, simply because the data are of mixed type. The same logic extends to Likelihood Ratio Test-based comparisons: models of one variable as a function of the others are not nested, and thus cannot be directly compared.

The assumption violations associated with both the eigen-based PCoA and GLM approaches motivated us to consider the underlying information directly through an information-theoretic lens.

### 1.2.1 Some definitions

We first consider discretized versions of all three variables, $(I, S, T)$, and for the time being, we take $S$ to be one-dimensional (i.e., we imagine individuals moving along a line). For reference and consistency, we define a few key information-theoretic terms below. Information for the various behavioural datasets shown in Figure 4 of the main text were quantified using the `infotheo` package in R [2].

**Definition 1** *The **entropy** of a discrete random varible $I$ with pmf $p(I)$ is*

$$H(I) = H(p(I)) = -\sum_{i \in \mathcal{I}} p(i) \log(p(i)) = E_{\mathcal{I}} \left[ \log \frac{1}{p(i)} \right] \tag{1}$$

In layman's terms, an observation's *entropy* is the amount of "surprisal" it carriers. In aggregate, a system's entropy is its unpredictability. Unusual observations (e.g., an individual observed far outside its typical home range) carry more entropy than more "typical" observations of the same individual.

**Definition 2** *The **joint entropy** $H(I, S, T)$ is*

$$H(I, S, T) = -\sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} p(i, s, t) \log(p(i, s, t) \tag{2}$$

In layman's terms, *joint entropy* is simply the entropy of a set of variables (as opposed to a single variable). The largest possible value of the joint entropy is the sum of the individual entropies of the contributing variables. This maximum is attained in the case where all variables are independent of one another.

**Definition 3** *The **conditional entropy** of $I$ given $(S, T)$ is*

$$H(I|(S, T)) = -\sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} p(i, s, t) \log \left( p(i|(s, t)) \right) \tag{3}$$

In layman's terms, conditional entropy measures the uncertainty remaining in $I$ after the values of variables $S$ and $T$ are known. Conditioning always reduces entropy, due to a relationship known as the "chain rule of conditional entropy", which states that if $X_1, X_2, ..., X_n$ are drawn according to $p(X_1, X_2, ..., X_n)$, then

$$H(X_1, X_2, ..., X_n) = \sum_{i=1}^{n} H(X_i | X_{i-1}, ..., X_1). \tag{4}$$

The conditional entropy of $H(I|(S,T)) = 0$ if $I$ is completely determined by $S$ and $T$. If $H(I|(S,T)) = H(I)$, then $I$ is completely independent of $S, T$.

**Definition 4** *The **mutual information** between variables X and Y, M(X;Y) is the reduction in entropy due to conditioning:*

$$I(Y;X) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right) \tag{5}$$

In essence, mutual information describes the mutual dependence between two variables. It tells us how much information we have about $X$ when $Y$ is known. This can alternatively be thought of as the Kullback-Leibler divergence between the joint distribution $p(X,Y)$ and the product distribution, $p(X)p(Y)$.

Mutual information is closely related to correlation, but unlike correlation, it is not limited to real values. It is constrained to be non-negative, and only equals zero in the case where the variables involved are independent of one another. We show direct application of mutual information to quantify information sharing among elements of $(I, S, T)$ in Figure 4 of the main text in the Section 1.3 of this supplementary document.

## 1.3 Quantifying mutual information in Figure 3

Figure 3 in the main text shows tripartite tagging networks for four different imagined societies. In all cases, there are six individuals observed at five points each in time, producing an data matrix of 30 observations (rows). The **M** matrix therefore requires columns for five individual indicator variables, two spatial coordinates (we imagine a row of patches here, so one spatial coordinate is always set to "1"), and one temporal coordinate, so that $\mathbf{M} = [I_1, I_2, I_3, I_4, I_5, S_1, S_2, T]$. Data matrices for all four panels are shown below. We used these matrices as a basis for calculating mutual information between pairs of variables in each case. Those mutual information values were used in turn to build the Venn diagrams in Figure 3.

$$\mathbf{3a} = \begin{bmatrix} 1&0&0&0&0&1&1&1 \\ 1&0&0&0&0&1&2&2 \\ 1&0&0&0&0&1&1&3 \\ 1&0&0&0&0&1&3&4 \\ 1&0&0&0&0&1&5&5 \\ 0&1&0&0&0&1&1&1 \\ 0&1&0&0&0&1&2&2 \\ 0&1&0&0&0&1&3&3 \\ 0&1&0&0&0&1&3&4 \\ 0&1&0&0&0&1&6&5 \\ 0&0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&2&2 \\ 0&0&1&0&0&1&5&3 \\ 0&0&1&0&0&1&3&4 \\ 0&0&1&0&0&1&1&5 \\ 0&0&0&1&0&1&5&1 \\ 0&0&0&1&0&1&5&2 \\ 0&0&0&1&0&1&5&3 \\ 0&0&0&1&0&1&5&4 \\ 0&0&0&1&0&1&3&5 \\ 0&0&0&0&1&1&5&1 \\ 0&0&0&0&1&1&6&2 \\ 0&0&0&0&1&1&5&3 \\ 0&0&0&0&1&1&1&4 \\ 0&0&0&0&1&1&3&5 \\ 0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&1&3&2 \\ 0&0&0&0&0&1&5&3 \\ 0&0&0&0&0&1&5&4 \\ 0&0&0&0&0&1&5&5 \end{bmatrix}; \mathbf{3b} = \begin{bmatrix} 1&0&0&0&0&1&1&1 \\ 1&0&0&0&0&1&3&2 \\ 1&0&0&0&0&1&2&3 \\ 1&0&0&0&0&1&6&4 \\ 1&0&0&0&0&1&4&5 \\ 0&1&0&0&0&1&1&1 \\ 0&1&0&0&0&1&3&2 \\ 0&1&0&0&0&1&2&3 \\ 0&1&0&0&0&1&6&4 \\ 0&1&0&0&0&1&4&5 \\ 0&0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&3&2 \\ 0&0&1&0&0&1&2&3 \\ 0&0&1&0&0&1&6&4 \\ 0&0&1&0&0&1&4&5 \\ 0&0&0&1&0&1&3&1 \\ 0&0&0&1&0&1&5&2 \\ 0&0&0&1&0&1&6&3 \\ 0&0&0&1&0&1&2&4 \\ 0&0&0&1&0&1&5&5 \\ 0&0&0&0&1&1&3&1 \\ 0&0&0&0&1&1&5&2 \\ 0&0&0&0&1&1&6&3 \\ 0&0&0&0&1&1&2&4 \\ 0&0&0&0&1&1&5&5 \\ 0&0&0&0&0&1&3&1 \\ 0&0&0&0&0&1&5&2 \\ 0&0&0&0&0&1&6&3 \\ 0&0&0&0&0&1&2&4 \\ 0&0&0&0&0&1&5&5 \end{bmatrix}; \mathbf{3c} = \begin{bmatrix} 1&0&0&0&0&1&1&1 \\ 1&0&0&0&0&1&2&2 \\ 1&0&0&0&0&1&3&3 \\ 1&0&0&0&0&1&4&4 \\ 1&0&0&0&0&1&5&5 \\ 0&1&0&0&0&1&1&1 \\ 0&1&0&0&0&1&2&2 \\ 0&1&0&0&0&1&3&3 \\ 0&1&0&0&0&1&4&4 \\ 0&1&0&0&0&1&5&5 \\ 0&0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&2&2 \\ 0&0&1&0&0&1&3&3 \\ 0&0&1&0&0&1&4&4 \\ 0&0&1&0&0&1&5&5 \\ 0&0&0&1&0&1&1&1 \\ 0&0&0&1&0&1&2&2 \\ 0&0&0&1&0&1&3&3 \\ 0&0&0&1&0&1&4&4 \\ 0&0&0&1&0&1&5&5 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&1&1&2&2 \\ 0&0&0&0&1&1&3&3 \\ 0&0&0&0&1&1&4&4 \\ 0&0&0&0&1&1&5&5 \\ 0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&1&2&2 \\ 0&0&0&0&0&1&3&3 \\ 0&0&0&0&0&1&4&4 \\ 0&0&0&0&0&1&5&5 \end{bmatrix} \mathbf{3d} = \begin{bmatrix} 1&0&0&0&0&1&1&1 \\ 1&0&0&0&0&1&1&2 \\ 1&0&0&0&0&1&1&3 \\ 1&0&0&0&0&1&1&4 \\ 1&0&0&0&0&1&1&5 \\ 0&1&0&0&0&1&2&1 \\ 0&1&0&0&0&1&2&2 \\ 0&1&0&0&0&1&2&3 \\ 0&1&0&0&0&1&2&4 \\ 0&1&0&0&0&1&2&5 \\ 0&0&1&0&0&1&3&1 \\ 0&0&1&0&0&1&3&2 \\ 0&0&1&0&0&1&3&3 \\ 0&0&1&0&0&1&3&4 \\ 0&0&1&0&0&1&3&5 \\ 0&0&0&1&0&1&4&1 \\ 0&0&0&1&0&1&4&2 \\ 0&0&0&1&0&1&4&3 \\ 0&0&0&1&0&1&4&4 \\ 0&0&0&1&0&1&4&5 \\ 0&0&0&0&1&1&5&1 \\ 0&0&0&0&1&1&5&2 \\ 0&0&0&0&1&1&5&3 \\ 0&0&0&0&1&1&5&4 \\ 0&0&0&0&1&1&5&5 \\ 0&0&0&0&0&1&6&1 \\ 0&0&0&0&0&1&6&2 \\ 0&0&0&0&0&1&6&3 \\ 0&0&0&0&0&1&6&4 \\ 0&0&0&0&0&1&6&5 \end{bmatrix}; \tag{6}$$

Code to build Figure 3 is included in the Code appendix to this supplement.

# 2 Applying marginal constraints using the tripartite model

In this section, we provide additional detail on the simulation protocols that we used to build constrained tripartite (and subsequently, unipartite) networks in section *Simulating tripartite networks subject to marginal constraints* in the main text.

A tripartite network can be written as an array of timestep-specific individual-by-site matrices (written $I \times S|T = t$). At its finest resolution this array is binary, with cells containing ones when a specific individual is present at a particular site during a particular time, and zeroes otherwise. The total number of ones in the array corresponds to the total number of edges in the tripartite network, and each unique configuration of a fixed number of ones within the array

describes one possible network. In the two examples below, we constrain one or several margins of this array, and simulate the position of a fixed number of ones subject to those marginal constraints. We then compare network metrics from simulated-but-constrained networks to network metrics calculated on simulated networks with the same total number of ones, but without marginal constraints.

## 2.1 Example 1

In Example 1, we imagine simulating a tripartite network subject to fixed constraints on individual home-range size and patch occupancy patterns, and using that network as a basis for generating *a priori* expectations about network modularity within the individual association network projection. In this example, those constraints were a marginal Gamma($\kappa = 5, \theta = 2$) distribution on home range size, and a marginal Gamma($\kappa = 0.5, \theta = 50$) distribution on patch occupancy frequencies.

### 2.1.1 Simulation protocol for Example 1

Prior to simulation, we generated data on two imagined variables, home range size and patch quality. Data were generated according to two different scenarios. In the first scenario, we considered a system where space use was highly overdispersed, so that most patches were used very rarely, while a few were used with extremely high frequency (Figure 4B in the main text). In the second scenario, we considered a system in which patches were relatively homogeneous in quality, so that all patches were used with similar frequency (Figure 4C in the main text).

In the first scenario, home range sizes were generated by a draw of 50 samples from a Gamma($\kappa = 5, \theta = 2$) distribution, with all values rounded up to the nearest integer. Patch quality values were generated by a single draw of size 30 from a Gamma ($\kappa = 0.5, \theta = 50$) distribution. In the second scenario, home range sizes were generated by a draw of 50 samples from a Gamma($\kappa = 2, \theta = 2$) distribution, with all values rounded up to the nearest integer. Patch quality values were generated by a single draw of size 30 from a Gamma ($\kappa = 5, \theta = 10$) distribution. Each value was then divided by the sum of all 30 values so that patch quality summed to 1 across all patches.

Following data generation, we simulated tripartite networks subject to these "observed" constrains using the protocol outlined here. An example of Step 6 is shown in detail in the subsequent section, labeled **Home range simulation example**.

1. Create $N_{\text{Individuals}} = 50$ individual nodes, $N_{\text{Patches}} = 30$ patch nodes, and $N_{\text{Timesteps}} = 40$ time nodes.
2. Assign $N_{\text{Timesteps}}$ "stubs" to each Individual node.
3. Draw a home range size for each individual (sampled with replacement from the observed distribution of home range sizes, and constrained so that the number of patches in each home range could not exceed $N_{\text{Timesteps}}$).

4. Draw a quality value for each patch (sampling with replacement from the observed distribution of patch qualities).

5. Identify the specific patches within each individual's home range.

   (a) Sum the home range sizes of all individuals to determine a total number of patches to assign to individual home ranges.

   (b) Draw that total number of patch labels from a multinomial distribution of patch qualities. This generates a "Full urn" of home range patches.

   (c) Select an individual at random, and work through the following steps:

      i. Draw a number of patches equal to the selected individual's home range size from the Full urn of home range patches.

      ii. Check to be certain that all patches drawn have unique labels. Redraw until a set of all unique patches is obtained.

      iii. Update the Full urn to a Reduced urn, by removing one patch of each type drawn by this individual. The next randomly selected individual will draw from the Reduced urn[1].

   (d) Select a new individual at random from the set of individuals whose home range patches have not been designated, and repeat this process.

6. Generate a number of visits to each patch within a home range, conditional on home range size and composition.

   (a) Identify a set of home range patches for each individual, conditional on individual home range size. This is done by drawing *Home Range Size*$_i$ patch labels without replacement from a Discrete Uniform(min = 1, max = $N_{\text{Patches}}$) distribution.

   (b) Require each individual to visit every patch within its home range at least once. Say for instance that the $i^{th}$ individual's home range encompasses $j$ patches. Then the remaining visits to be allocated for Individual $i$ would be

$$\textit{Remaining visits}_i = N_{\text{Timesteps}} - j$$

   (c) Allocate the remaining visits to patches as follows.

      i. Determine the number of additional timesteps to be allocated across all individuals,

$$N_{\text{Allocation total}} = \sum_{i \in \mathbb{I}} \textit{Remaining visits}_i$$

      where $\mathbb{I}$ is the set of all individuals.

[1] If it is not possible to draw enough uniquely labeled patches, draw patches directly from the original Full urn.

ii. Generate a single draw of size $N_{\text{Allocation total}}$ patch labels from a multinomial distribution with probabilities equal to the probabilities in the patch quality distribution. This multinomial draw produces a "Full urn", containing $N_{\text{Allocation total}}$ patches distributed according to the patch quality distribution.

iii. Randomly select a focal individual.

iv. Identify an "Individual-specific urn" for the focal individual by subsetting the Full urn down to only those patches in the Full urn that are also contained in the focal individual's home range.

v. Draw the appropriate number of visits ($N_{\text{Timesteps}}$ − (Focal individual's home range size)) from the Individual-specific urn without replacement.[2] The focal individual's total space use is described by this set of patches, combined with one additional visit to each patch in the focal individual's home range.

vi. Remove the patch visits for the focal individual from the Full urn to produce a Reduced urn.

vii. Remove the focal individual from the set of possible animals, select a new individual from this reduced set at random, and repeat the process, this time using the Reduced urn in place of the Full urn.

viii. After identifying a set of patches for the new individual, remove that individual from the set of individuals, and remove the new individual's patches from the Reduced urn to create a Reduced urn'. Randomly select a third individual.

ix. Continue this process until all individuals have been chosen.

We include an example of this simulation process in the next section, **Home-range simulation example**.

7. Once a set of 40 patch visits is identified for all individuals, randomly assign a timesteps to each patch visit by each individual (now sampling without replacement from the set of timesteps).

8. Bind Individual Identity, Patch, and Time vectors together into a data frame. Build an "Aggregation" variable by pasting Patch and Time labels together for each row. Use the Individual Identity and Aggregation variables to construct a bipartite individual-aggregation network. Project that network down to the corresponding individual association network.

9. Use a four-step walk trap algorithm to calculate modularity on the individual association network, and store the modularity value produced. Modularity ($Q$) was calculated as:

$$Q = \frac{1}{(2m)} \sum_{ij} \left[ A_{ij} - \frac{k_i k_j}{(2m)} \right] \delta\left(c_I, c_j\right) \tag{7}$$

where $m$ equals the total number of edges in the graph, $i$ and $j$ index particular nodes with degrees $k_i$ and

[2] If the size of the Individual-specific urn is less than the number of visits, expand the Individual-specific urn as follows. Retain its original contents, and add to them a sample of patches from the Individual-specific urn drawn with replacement, that is equal in size to the additional number of patches required.

$k_j$ respectively. $A_{ij}$ is the adjacency matrix, $c_i$ and $c_j$ identify the graph components in which nodes $i$ and $j$ are members, with $\delta(c_i, c_j)$ describing agreement between those components (i.e., $\delta(c_i, c_j) = 1$ if $c_i = c_j$, and 0 otherwise).

### 2.1.2 Home-range simulation example

Imagine a population of 3 individuals ($I_1, I_2, I_3$), occupying 4 patches (A, B, C, D) over 5 Timesteps ($T_1, T_2, T_3, T_4, T_5$). After Steps 3 and 5a, Individuals are determined to have the following home-ranges:

- $I_1$ has home-range $\{A, B\}$
- $I_2$ has home-range $\{A, C\}$
- $I_3$ has home-range $\{B, C\}$.

The relative patch quality distribution is $Pr[A] = .15, Pr[B] = .25, Pr[C] = .6$.

$I_1$ must visit each patch in its home-range at least once, so it must spend at least one timestep in $A$ and one timestep in $B$; $I_2$ must spend at leat one timestep in $A$ and one timestep in $C$; and $I_3$ must spend at least one timestep in $B$ and one timestep in $C$.

Step i: For each individual, two timesteps must be used to cover the individual's home-range, leaving three timesteps that can be allocated to patches at random, according to the patch occupancy distribution. This means that $N_{\text{Allocation total}} = 9$.

Step ii: Generate a draw of size 9 from a Multinomial$(.15, .25, .6)$. In this example, assume this draw produces the following Full urn:

$$\text{Full urn} = \{A, A, B, B, B, C, C, C, C\}$$

Step iii: Randomly select an individual, in this case, Individual 2.

Step iv: Generate an Individual 2-specific urn by limiting the Full urn to only those patches that are in Individual 2's home range, $\{A, C\}$. This produces an Individual 2-specific urn of:

$$\text{Individual 2-specific urn} = \{A, A, C, C, C, C\}$$

Step v: Since Individual 2 must spend one timestep in A, and one timestep in C, and we have 5 timesteps, 3 timesteps remain to be assigned. Identify these patches by drawing 3 elements without replacement from the Individual 2-specific urn. Presume these elements are $\{C, C, C\}$. All of Individual 2's patch visits over the 5 timesteps are then

described by these three elements, plus an additional visit to A and C:

$$\text{Individual 2's patch visits} = \{A, C, C, C, C\}$$

Step vi: Produce the Reduced urn by removing the patches drawn by Individual 2 from the Full urn:

$$\text{Reduced urn} = \{A, A, B, B, B, C, C, C, C\} - \{C, C, C\} = \{A, A, B, B, B, C\}$$

Step vii: Select a new individual from the remaining individuals (here, $I_1$ and $I_3$; presume in this case that we drew $I_3$, with home-range $\{B, C\}$. The Individual 3-specific urn consists of all elements in the Reduced urn that are also in Individual 3's home range:

$$\text{Individual 3-specific urn} = \{B, B, B, C\}$$

Draw a sample of size 3 without replacement from this Individual 3-specific urn. In this case, say this sample consists of $\{B, B, C\}$. Add one additional visit to each patch in Individual 3's home-range to get Individual 3's patch visits:

$$\text{Individual 3's patch visits} = \{B, B, B, C, C\}$$

Step viii: Remove the sampled patches from the Reduced urn to get a Reduced urn':

$$\text{Reduced urn'} = \{A, A, B, B, B, C\} - \{B, B, C\} = \{A, A, B\}$$

Step ix: Remove $I_3$ from the set of remaining individuals. Draw another individual from that set (in this case, this must be $I_1$). Generate an Individual 1-specific urn consisting of all elements in Reduced urn' that are also in $I_1$'s home-range:

$$\text{Individual 1-specific urn} = \{A, A, B\}$$

Sample without replacement from the Individual 1-specific urn (here, getting the set $\{A, A, B\}$). Add one additional visit to each patch in Individual 1's home range to get Individual 1's patch visits:

$$\text{Individual 1's patch visits} = \{A, A, A, B, B\}$$

.

If there were additional individuals in the population, we would then generate a new Reduced urn", sample a new individual, and continue with this protocol until all individuals had been assigned patch visits.

### 2.1.3 Validating the distributional form for relative patch qualities in Example 1

Drawing the patches included in each individual's home range from a discrete uniform could potentially lead the realized patch quality distribution to depart from the ones specified at the simulation's outset. In particular, this method will systematically increase realized use of patches of very low quality, and at the same time systematically decrease use of patches of very high quality. We assessed the size of this departure by comparing emergent patch quality distributions after applying constraints to the input distribution of patch qualities. Specifically, we compared maximum likelihood fits of the realized patch quality distribution to the parameters specified at the simulation's output.

We found that the distribution was only minimally changed. We initially drew patch quality rates from a Gamma($\alpha = 0.5$, $\theta = 50$) distribution. After determining patch occupancies as constrained by home range size in Step 5, we estimated the Gamma distribution parameters under the observed patch occupancy rates. 95% of fitted values of $\alpha$ fell between 0.45 and 0.55, while 95% of the fitted values for $\theta$ fell between 42.40 and 47.00. As a consequence, while Steps 4 and 5 undoubtedly did alter the shape of the patch occupancy distribution away from its specified structure, this change was qualitatively quite small. Maximum likelihood estimates for the Gamma distributions were generated using the `fitdist()` function in `R`'s `fitdistrplus` package[3].

## 2.2 Example 2

In Example 2, our intent was to systematically vary the structure of the group size distribution (in habitat of heterogeneous quality), and observe how changes in that structure shifted the distributions of other marginal quantities.

### 2.2.1 Data generation

Prior to simulation, we generated mock data on both aggregation size and patch quality. We generated tuneable aggregation size distributions by systematically varying the parameters of a Gamma($\alpha$, $\theta$) distribution to generate an *a priori* set of aggregation sizes ($\alpha$ ranged systematically on the log-scale from 0.5 to 50; $\theta$ ranged from 2 to 10). We introduced heterogeneity in patch quality by requiring patches to be occupied in a specific order (so that in timesteps with 5 aggregations, the 5 top-ranked patches were occupied, but when only 3 aggregations were present, the 4th and 5th-ranked patches were left empty). $N_{\text{Individuals}}$ was set to 50, $N_{\text{Patches}}$ to 40, and $N_{\text{Timesteps}}$ to 40 throughout.

### 2.2.2 Simulating aggregation size for Example 2

In order to assign a fixed number of individuals to a variable number of aggregations in accordance with some user-defined aggregation size distribution, we needed to place a probability distribution on all possible partitions of the individuals into aggregations.

We first considered drawing aggregation memberships from a multinomial distribution, with the probabilities in the multinomial designed to capture the aggregation size distribution. This approach was problematic in two ways. First, in order to conduct the multinomial draw, we needed to specify how many aggregations were present, so that we could define the set of possible outcomes, and the length of the probability vector. This was problematic, since we wanted to allow number of aggregations to emerge from the simulation. Second, even if we could appropriately define the number of groups present (i.e., the multinomial's "states"), calculating probabilities associated with each group so that the expected distribution of individuals across groups matched our desired expectation was not trivial.

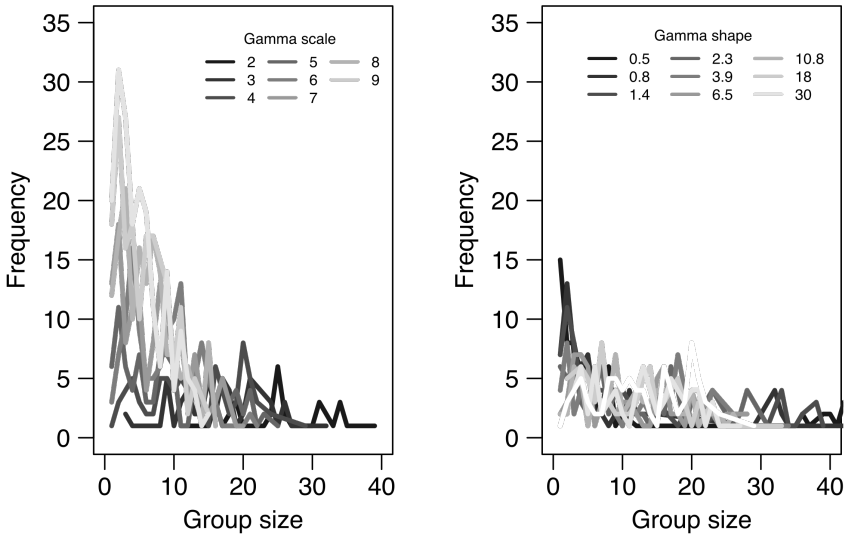


Figure 1: Aggregation size distributions generated by our simulation procedure using Gamma distributions with variable scale ("$\theta$"; left) and shape ("$\alpha$"; right) parameter values.

Given the problems with the multinomial approach, we switched to a randomization scheme inspired by point processes. We simulated aggregations uniquely for every day. Our strategy was to first generate $N_{\text{Individuals}}$ points from a Gamma($\alpha$, $\theta$) distribution, where $\theta$ is a scale (as opposed to a rate) parameter. This gave us $N_{\text{Individuals}}$ points scattered (non-homogeneously) along the positive real line. We then cut the line into $\theta$ bins of equal length. All individuals within a bin were then assigned to the same aggregation. Altering $\theta$ allowed us to systematically alter the aggregation size distribution. Aggregation size distributions under various values of $\theta$ (and $\alpha$) are shown in Figure 1.

#### 2.2.3 Simulation protocol for Example 2

1. Create $N_{\text{Individuals}}$ individual nodes, $N_{\text{Patches}}$ patch nodes, and $N_{\text{Timesteps}}$ time nodes.

2. Assign $N_{\text{Timesteps}}$ "stubs" to each Individual node

3. Generate a set of aggregations for each timestep, conditional on the number of individuals present at that

timestep:

(a) Draw $N_{\text{Individuals}}$ points from a Gamma($\alpha$, $\theta$) distribution. This produces a set of $N_{\text{Individuals}}$ points stochastically positioned along a continuous line, with clustering dependent on the Gamma parameters.

(b) Split the points into bins with width $\frac{1}{\theta}$, and assign all individuals in the same bin to a common aggregation. The number of aggregations ($N_{\text{Aggregations}}$) is the equal to the number of bins (see Section 3.2 above).

(c) Draw $N_{\text{Aggregations}}$ patches without replacement from the set of $N_{\text{Patches}}$. Allow for heterogeneous patch quality by weighting the probability that each patch is drawn.

4. Bind Identity, Patch, and Time vectors together into a data frame. Build an "Aggregation" variable by pasting Patch and Time labels together for each row. Use the Identity and Aggregation variables to construct a bipartite individual-aggregation network. Project that network down to the corresponding individual association network.

5. Use a four-step walk trap algorithm to calculate and record modularity of the individual association network. Calculate individual home range size by summing the total number of patches visited, and record mean home range size. Calculate the size of each aggregation and record mean aggregation size.

# References


[1] Pierre Legendre and Loic FJ Legendre. *Numerical ecology*, volume 24. Elsevier, 2012.

[2] Patrick E Meyer. infotheo: Information-theoretic measures, 2014. *R package version*, 1(0).

[3] Marie Laure Delignette-Muller, Christophe Dutang, et al. fitdistrplus: An r package for fitting distributions. *Journal of Statistical Software*, 64(4):1–34, 2015.


# Code

The `R` code below was used to quantify mutual information and construct the Venn diagrams in Figure 3.

```
# Require infotheo package for calculating mutual information
require(infotheo)


# construct desired I,S,T dataframe, with I coded as a single, categorical variable.


# calculate all pairwise mutual informations among (I,S,T) elements using mutinformation function.
```

```
mia <- mutinformation(Obs_4a_df)
mib <- mutinformation(Obs_4b_df)
mic <- mutinformation(Obs_4c_df)
mid <- mutinformation(Obs_4d_df)

# Require venneuler package for plotting
require(venneuler)

# Set up plot layout
par(mfrow = c(2, 2))

# Venn diagram for Figure 4a

venn.a <- venneuler(c("I" = mia[1, 1] - sum(mia[-1, 1]),
                      "S" = mia[3, 3] - sum(mia[-3, 3]),
                      "T" = mia[4, 4] - sum(mia[-4, 4]),
                      "I&S" = mia[1, 3],
                      "I&T" = mia[1, 4],
                      "S&T" = mia[3, 4]))
plot(venn.a)

# Venn diagram for Figure 4b

venn.b <- venneuler(c("I" = mib[1, 1] - sum(mib[-1, 1]),
                      "S" = mib[3, 3] - sum(mib[-3, 3]),
                      "T" = mib[4, 4] - sum(mib[-4, 4]),
                      "I&S" = mib[1, 3],
                      "I&T" = mib[1, 4],
                      "S&T" = mib[3, 4]))
plot(venn.b)

# Venn diagram for Figure 4c
```

```
venn.c <- venneuler(c("I" = mic[1, 1] - sum(mic[-1, 1]),
                      "S" = mic[3, 3] - sum(mic[-3, 3]),
                      "T" = mic[4, 4] - sum(mic[-4, 4]),
                      "I&S" = mic[1, 3],
                      "I&T" = mic[1, 4],
                      "S&T" = mic[3, 4]))
plot(venn.c)

# Venn diagram for Figure 4d
venn.d <- venneuler(c("I" = mid[1, 1] - sum(mid[-1, 1]),
                      "S" = mid[3, 3] - sum(mid[-3, 3]),
                      "T" = mid[4, 4] - sum(mid[-4, 4]),
                      "I&S" = mid[1, 3],
                      "I&T" = mid[1, 4],
                      "S&T" = mid[3, 4]))
plot(venn.d)
```