**Electronic supplementary material**

A place for everything and everything in its place: spatial organization of individuals on nests of the primitively eusocial wasp *Ropalidia marginata*

Nitika Sharma\*, Raghavendra Gadagkar

\*Corresponding author - nitikasharma@iisc.ac.in

Proceedings of the Royal Society B

doi: 10.1098/rspb.2019.1212

**Contents Page number**

Materials and methods 1

Tables 4

Figures 8

References 15

R codes 17

**Materials and methods**

*Behavioural data collection*

We collected quantitative behavioural data (for detailed explanation of the behavioural repertoire of *Ropalidia marginata*, please see [1]) from video recordings on each of six nests. Each nest was video recorded from 8:00 A.M. to 6:00 P.M. for three consecutive days. From the video replay, long-duration behaviours of all wasps were recorded during scans and short-duration behaviours were recorded during all occurrence sessions (AOS), both of which lasted five minutes with a minute long break in between. The sequence of scans and AOS sessions was randomly intermingled. At the end, 150 scans and 150 AOS respectively, were obtained for each nest. The proportions of time spent by each wasp performing each behaviour, were calculated from the scans and the frequencies per hour with which each wasp performed each rare behaviour, were calculated from the AOS sessions.

*Core area delineation*

We used the package adehabitatHR [2] in RStudio (Version 1.1.453) for creating core area maps based on the relocation data of each wasp using kernel density estimation (KDE). Core areas are any areas in the home range of an animal that are used more intensively than others [3] and are often denoted by the region where the animal can be located with 50% probability [4]. Kernel density estimation developed by Worton [5] and also used by us, is the most accepted and widely used method for core area delineation because it is free of the parametric assumptions and is less sensitive to outliers [6]. Core area maps were created for each of the 155 wasps across 6 nests and the area of this 50% kernel was also recorded. The smoothening parameter used was ‘href’, which is more suitable than LSCV for our data as LSCV fails to converge when the animal uses identical locations repeatedly. According to Seaman and Powell [2,7] "This is a difficult problem that has not been worked out by statistical theoreticians, so no definitive response is available at this time".

*Role of brood*

We used compositional analysis used for habitat selection studies from the adehabitatHS package [8] in RStudio (Version 1.1.453). The available and used larval cells were calculated for each of the 101 wasps that showed spatial fidelity. The cells ‘available’ to a wasp were the proportion of cells on a nest that were occupied by larvae while that ‘used’ by a wasp were the proportion of cells within the core area that were occupied by larvae. We therefore analysed if the used proportion of larval cells was greater than what was available. The spatial distribution of adult wasps on the nests with respect to the spatial distribution of larvae on their nests indicated that adult wasps did not select larvae containing cells more often than non-larval cells (compositional analysis, λ = 0.98, p = 0.15).

To assess if the observed location of the core area had significantly more larvae than five randomly located core areas of the same size, we performed a mixed conditional logistic regression [9] using the coxme package [10] in Rstudio (Version 1.1.453). The observed core area for each wasp was assigned 1 for occupancy while the random core areas were assigned 0 to indicate that these locations were not occupied.

Each brood stage in each of the six nests was spatially located in the centre of the cell containing it, using ImageJ. Clark and Evans R measure (spatstat package [11] in RStudio Version 1.1.453) was used to check statistically, if larval cells are clustered on the nest surface. This method estimates the aggregation between spatial points based on the nearest neighbour distances in the observed spatial pattern and compares it to randomly located spatial points [12].

*Ovarian index*

The ovarian index of each of the wasps collected from the nests on the fourth day after the three days of video-recording was over. Each wasp was dissected under a microscope under 10X resolution and six parameters of ovaries of each wasp were calculated. These parameters were: i. No. of mature eggs ii. Width of the proximal oocyte iii. Length of the proximal oocyte iv. Total number of oocytes v. Number of oocytes with yolk vi. Number of resorbing oocytes. All these six measures were separately recorded for each of the six ovarioles of a wasp. The ovarian development index was calculated as the first principal component calculated after performing a principal components analysis on these six parameters.

Generalized linear mixed effects models and mixed effects models were run using lme4 package [13] in RStudio (Version 1.1.453) [14].

**Tables**

**Table S1**: Generalized mixed effects model was used to understand if the behavioural roles of wasps predicted the probability that wasps show spatial fidelity. Nest ID was used as the random effect. One, two and three asterisks (\*) indicate statistical significance at α = 0.05, α = 0.01 and α = 0.001.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Estimate | Bootstrapped  CI (2.5%) | Bootstrapped  CI (97.5%) | χ2 | df | Randomized p value | R2m | R2c |
| (Intercept) | 3.21 | 1.81 | 4.51 |  |  | 0.001 |  |  |
| **Feeding Larvae** | **-0.45** | **-1.27** | **- 0.03** | **4.52** | **1** | **0.04\*** | **0.05** | **0.03** |
| Maintenance | -0.31 | -0.61 | 0.14 | 0.93 | 1 | 0.34 | 0.01 | 0.01 |
| **Subordinate**  **behaviour** | **-0.61** | **- 1.57** | **- 0.08** | **8.9** | **1** | **0.01\*** | **0.01** | **0.08** |
| Dominance  behaviour | -0.19 | - 0.4 | 0.1 | 2.32 | 1 | 0.15 | 0.02 | 0.02 |
| **Foraging duration** | **-3.17** | **- 4.2** | **- 1.8** | **10.62** | **1** | **0.002 \*\*** | **0.11** | **0.11** |

**Table S2**: Feeding larvae frequency and subordinate behaviour frequency were found to significantly correlate with the size of core areas of wasps (normalized by nest size). A linear mixed effects model was used with nest ID as random effect. One, two and three asterisks (\*) indicate statistical significance at α = 0.05, α = 0.01 and α = 0.001.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Estimate | Bootstrapped CI (2.5%) | Bootstrapped CI (97.5%) | χ2 | df | Randomized  p value | R2m | R2c |
| (Intercept) | 18.57 | 15.12 | 24.43 |  |  |  |  |  |
| **Feeding Larvae** | **2.55** | **0.09** | **4.14** | **4.92** | **1** | **0.03\*** | **0.04** | **0.07** |
| Maintenance | 1.1 | -3.64 | 3.7 | 0.63 | 1 | 0.43 | 0.005 | 0.01 |
| **Subordinate behaviour** | **3.8** | **2.23** | **8.58** | **8.92** | **1** | **0.003 \*\*** | **0.07** | **0.05** |
| Dominance behaviour | 0.47 | -0.31 | 2.15 | 0.45 | 1 | 0.5 | 0.001 | -0.001 |
| Foraging  duration | 4.58 | -3.14 | 9.2 | 1.35 | 1 | 0.24 | 0.01 | 0.01 |

**Table S3:** Feeding larvae frequency and foraging duration were found to strongly correlate with the distance of a wasp from the colony centre. Subordinate behaviour was weakly correlated. A linear mixed effects model with nest ID as random effect was used for bootstrapping and randomization. One, two and three asterisks (\*) indicate statistical significance at α = 0.05, α = 0.01 and α = 0.001.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Estimate | Bootstrapped  CI (2.5%) | Bootstrapped  CI (97.5%) | χ2 | Randomized  p value | R2 m | R2 c |
| (Intercept) | 0.66 |  |  |  |  |  |  |
| **Feeding Larvae** | **-0.12** | **-0.22** | **-0.04** | **5.57** | **0.02 \*** | **0.03** | **0.02** |
| Maintenance | 0.08 | 0.003 | 0.2 | 1.32 | 0.3 | 0.006 | 0.006 |
| Subordinate behaviour | -0.09 | -0.2 | -0.014 | 3.48 | 0.055. | 0.02 | 0.009 |
| Dominance Behaviour | 0.04 | -0.04 | 0.2 | 1.52 | 0.25 | 0.007 | 0.007 |
| **Foraging**  **Duration** | **0.63** | **0.25** | **1.04** | **8.86** | **0.003 \*\*** | **0.05** | **0.05** |

**Table S4:** Clark-Evans R measure for the centroids of all wasps’ spatial distribution on the nest. A value R>1 suggests ordering, while R<1 suggests clustering.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | NEST V57 | NEST V82 | NEST V72 | NEST V87 | NEST V99 | NEST V14 |
| R {clustered if (R < 1)} | | | | | | |
| Adults | R = 0.51,  p <<0.001 | R = 0.49,  p <<0.001 | R = 0.69,  p <<0.001 | R = 0.68,  p <<0.001 | R = 0.55,  p <<0.001 | R = 0.42,  p <<0.001 |

**Table S5:** Frequency of snatching things had strong positive detectable relationship with the proportion of spatial overlap of core areas between pairs of wasps. Frequency of soliciting was weakly correlated with it. Linear mixed effects model was used with nest ID as random effect for bootstrapping and randomization. One, two and three asterisks (\*) indicate statistical significance at α = 0.05, α = 0.01 and α = 0.001.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Estimate | Bootstrapped CI (2.5%) | Bootstrapped CI (97.5%) | df | Pr(>Χ2) |
| (Intercept) | - 0.63 | - 0.66 | - 0.3 |  |  |
| Dominance Behaviour | 0.42 | -0.09 | 0.45 | 1 | 0.24 |
| **Exchange of regurgitated food (SC)** | **0.45** | **0.08** | **1.001** | **1** | **0.004 \*\*** |
| **Exchange of solid food (ST)** | **1.68** | **0.33** | **2.02** | **1** | **<0.001 \*\*\*** |

**Table S6**: Clark-Evans R measure for the different brood classes of each nest. A value R>1 suggests ordering, while R<1 suggests clustering.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | NEST V57 | NEST V82 | NEST V72 | NEST V87 | NEST V99 | NEST V14 |
|  | R {clustered if (R < 1)} | | | | | |
| Empty  Cells | R = 1.09,  p = 0.80 | R = 1.17,  p = 0.94 | R = 1.12,  p = 0.92 | R = Inf,  p = 1 | R = 1.16,  p = 0.97 | R = 2.02,  p = 1 |
| Eggs | R = 1.27,  p = 1 | R = 1.29,  p = 1 | R = 1.26,  p = 1 | R = 1.21,  p = 1 | R = 1.23,  p = 0.99 | R = 1.37,  p = 1 |
| Larvae | R = 1.15,  p = 0.96 | R = 1.18,  p = 0.98 | R = 1.21,  p = 1 | R = 1.30,  p = 1 | R = 1.10,  p = 0.93 | R = 1.17,  p = 0.96 |
| Pupae | R = 0.77,  p = 0.09 | R = 1.23,  p = 1 | R = 0.93,  p = 0.38 | R = 0.99,  p = 0.44 | R = 0.31,  p = 1 | R = 1.22,  p = 0.90 |

**Table S7:** Foraging duration was found to be strongly positively correlated with the average instantaneous Euclidean distance of worker from the queen. Frequency of subordinate behaviour was found to have a weak positive detectable correlation. Linear mixed effects model with nest ID as random effect was used for bootstrapping and randomization. One, two and three asterisks (\*) indicate statistical significance at α = 0.05, α = 0.01 and α = 0.001.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Estimate | Bootstrapped  CI (2.5%) | Bootstrapped  CI (97.5%) | χ2 | df | Randomized  p values | R2 m | R2 c |
| (Intercept) | 2.59 | 2.4 | 2.7 |  |  | 1 |  |  |
| Feeding Larvae | 0.002 | -0.02 | 0.08 | 0.003 | 1 | 0.94 | -0.0003 | -0.0007 |
| Maintenance | 0.06 | 0.02 | 0.1 | 2.2 | 1 | 0.14 | 0.007 | 0.006 |
| Subordinate behaviour | 0.07 | 0.01 | 0.1 | 3.6 | 1 | 0.06 | 0.01 | 0.02 |
| Dominance Behaviour | 0.016 | -0.02 | 0.04 | 0.94 | 1 | 0.4 | 0.003 | 0.006 |
| **Foraging** | **0.41** | **0.2** | **0.5** | **11.94** | **1** | **<0.001\*\*\*** | **0.04** | **0.07** |

**Table S8:** The queen had significantly fewer partners involved in trophallaxis (exchange of regurgitated food among adults) as compared to PQ and other workers while PQ had significantly more. Generalized linear mixed effects model was used with Poisson error family at α = 0.05.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Estimate | Std. error | CI  (2.5%) | CI  (97.5%) | z | p |
| Intercept (worker) | 2.3 | 0.1 | 2.05 | 2.5 | 23.7 | <0.001\*\*\* |
| Queen | - 0.94 | 0.2 | -1.42 | -0.53 | -4.18 | <0.001\*\*\* |
| Potential queen | 0.78 | 0.09 | 0.6 | 0.9 | 8.5 | <0.001\*\*\* |

**Figures**

****

**Fig. S1:** Cartoon representation of spatial overlap between pairs of wasps measures as the black region of overlap quantified asymmetrically as proportion of wasp A’s core area (red circle) overlapped by wasp B’s core area (orange circle) and vice versa. This calculation was made using adehabitatHR package [2] in RStudio (Version 1.1453).

**G:\Dropbox\For ProcB paper\Supp figs\5circles300.tif**

**Fig. S2**: A pictorial representation explaining how random core areas were placed within the nest boundary such that they matched the observed core area of a wasp in its size but not its location or shape.



**Fig. S3:** Cartoon representation of the method used for quantifying distance of a wasp from colony centre. The centroid of pooled spatial locations of all wasps over all days was considered the colony centre. The Euclidean distance of the centroid of each wasp’s spatial spread from the colony centre was measured as the wasp’s location with respect to colony centre.



**Fig. S4:** Left panel: Temporal variation over the 3 days in core area use of the queen (Q), potential queen (PQ) and a worker (-D) within nest v57. Right panel: Spatial overlap between core areas of the same wasp across days was significantly higher than spatial overlap between core areas of different wasps (Wilcoxon rank sum test, W = 272550, p < 0.001\*\*\*).

G:\Dropbox\For ProcB paper\resubmission\figures\ed_centre\FL300.tifG:\Dropbox\For ProcB paper\resubmission\figures\ed_centre\FG300.tif

**Fig. S5**: Wasps that frequently fed larvae were located closer to the colony centre while foragers were located farther away. Linear mixed effects model was used with nest as random effect at α = 0.05.

G:\Dropbox\For ProcB paper\Supp figs\S3a300.tif

**Fig. S6a**: Wasps that were active i.e. had bigger step length were located closer to the colony centre. Linear mixed effects model was used with nest as random effect (estimate = -0.39, bootstrapped CI = -0.82 to 0.008, p value = 0.001\*\*).

G:\Dropbox\For ProcB paper\Supp figs\S3b300.tif

**Fig. S6b**: Wasps that had a bigger step length tended to have bigger core areas (linear mixed effects model: estimate = 17.82, bootstrapped CI = 13.26 to 21.66, p < 0.001 \*\*\*).

G:\Dropbox\For ProcB paper\resubmission\figures\Ovarian index\with UD50.tiff

**Fig. S7a**: Wasps that had better developed ovaries occupied bigger core areas (Linear mixed effects model: est = 1.3, χ2 = 10.8, df = 1, p = 0.001\*\*).

G:\Dropbox\For ProcB paper\resubmission\figures\Ovarian index\step length.tiff

**Fig. S7b**: Wasps that had better developed ovaries had higher activity levels (Linear mixed effects model: est = 0.1, χ2 = 22.7, df = 1, p < 0.001\*\*\*).



**Fig. S8**: The queen in each of the six nests displayed significantly higher avoidance behaviour (deflection away from an approaching nest mate) than any other worker (Mixed effects logistic regression; z = 2.91, p = 0.003\*\*).

C:\Users\Nitika Sharma\Dropbox\thesis\Manuscript 1\SC_Q300.tif

**Fig. S9**: Queens had significantly fewer partners for exchanging regurgitated food while PQ had significantly higher than other workers. Generalized linear mixed effects model was used with with nest as random effect and Poisson error family at α = 0.05.

G:\Dropbox\For ProcB paper\resubmission\figures\org imm\avgdeg_UD50300.tif

**Fig. S10**: The average number of social partners a wasp interacts with is positively correlated with the size of its normalized core area (Linear mixed effects model; estimate – 0.22, bootstrapped CI - 0.13-0.36, χ2 = 16.4, p <0.001\*\*\*).

**References**

1. Gadagkar R. The social biology of Ropalidia marginata: toward understanding the evolution of eusociality. Harvard University Press; 2001 Jun 30.
2. Calenge C. Home range estimation in R: the adehabitatHR package. Office national de la classe et de la faune sauvage: Saint Benoist, Auffargis, France. 2011 Mar.
3. Burt WH. Territoriality and home range concepts as applied to mammals. Journal of mammalogy. 1943 Aug 1;24(3):346-52.
4. Laver PN, Kelly MJ. A critical review of home range studies. The Journal of Wildlife Management. 2008 Jan;72(1):290-8.
5. Worton BJ. Kernel methods for estimating the utilization distribution in home‐range studies. Ecology. 1989 Feb;70(1):164-8.
6. Bingham BB, Noon BR. Mitigation of Habitat “Take”: Application to Habitat Conservation Planning: Aplicación de la Mitigación de la “Toma” de Hábitat en la Planeación de Conservación de Hábitat. Conservation Biology. 1997 Feb 17;11(1):127-39.
7. Seaman DE, Griffith B, Powell RA. KERNELHR: a program for estimating animal home ranges. Wildlife Society Bulletin. 1998 Apr 1:95-100.
8. Calenge C. Exploratory Analysis of the Habitat Selection by the Wildlife in R: the adehabitatHS Package. Office national de la chasse et de la faune sauvage, Saint Benoist, France. 2011 Feb.
9. Duchesne T, Fortin D, Courbin N. Mixed conditional logistic regression for habitat selection studies. Journal of Animal Ecology. 2010 May;79(3):548-55.
10. Therneau T. Coxme: Mixed Effects Cox Models, R Package Version 2.2-3, 2012.
11. Baddeley A, Turner R. Spatstat: an R package for analyzing spatial point patterns. Journal of statistical software. 2005 Jan 26;12(6):1-42.
12. Clark PJ, Evans FC. Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology. 1954 Oct;35(4):445-53.
13. Bates D, Maechler M, Bolker B, Walker S. lme4: Linear mixed-effects models using Eigen and S4. R package version 1.1–7 [Internet]. 2014.
14. R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

**R codes**

1. **Core area delineation**

all<-read.csv("**v57\_day123\_raw**.csv")

tail(all)

dat<- all[!(is.na(all$X)),]

length(levels(dat$id))

levels(dat$id)

length(dat[,4])

## pruning off-nest pts

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

datx<-dat[,4]

daty<-dat[,5]

pol.x<-xnest

pol.y<-ynest

library(sp)

overall.pts<-point.in.polygon(datx,daty, pol.x, pol.y, mode.checked=FALSE)

dataset<-dat[,c(1,2,3,4,5)]

head(dataset)

nrow(dataset)

include.pts<-cbind(dataset,overall.pts)

include.pts<-as.data.frame(include.pts)

pt.ins1<-include.pts[ which(overall.pts=='1'),] #pts that fall inside the polygon

pt.ins3<-include.pts[ which(overall.pts=='3'),]

#head(pt.ins3)

sub\_dat<-rbind(pt.ins1,pt.ins3) #row bind for all x,y that are within nest polygon

par(mfrow=c(1,1))

plot(sub\_dat$X,sub\_dat$Y) #successfully subsetted to only the ones that lie in the cell

v57\_centreX<-mean(sub\_dat$X)

v57\_centreY<-mean(sub\_dat$Y)

points(v57\_centreX,v57\_centreY, pch="+", col="red")

library(adehabitatHR)

xtot<-sub\_dat$X

ytot<-sub\_dat$Y

xytot<-cbind(xtot,ytot)

dftot<-data.frame(X=xytot[,1],Y=xytot[,2])

dfsp <- SpatialPoints(xytot)

class(dfsp)

idsp<-data.frame(sub\_dat[,2])

sppt<-data.frame(dfsp)

coordinates(idsp)<-dftot

##for the entire nest

kernel\_tot<-kernelUD(idsp[,1], h = "href", grid = 64, hlim=c(0,3),

same4all = TRUE, kern = c("epa"), extent = 0.5)

kernelUD(idsp[,1])

image(kernel\_tot)

#####kernelUD output- IMG 1 #####

core\_tot <- getverticeshr(kernel\_tot, percent=50)

class(core\_tot)

as.data.frame(core\_tot)

plot(core\_tot, display="filled.contour2", xlab="X", ylab="Y", add=T, lwd=3)

points(dfsp, cex=0.2, pch=16) ##plotting datapts

as.data.frame(core\_tot)

ii\_tot <- kernel.area(kernel\_tot, percent=c(50, 95))

ii\_tot

###volume of kernel- IMG 2 #####

vol\_tot\_ud <- getvolumeUD(kernel\_tot)

image(vol\_tot\_ud[[1]],cont=50)

title("Output of getvolumeUD-total for nest v57")

xyzv <- as.image.SpatialGridDataFrame(vol\_tot\_ud[[1]])

contour(xyzv, add=TRUE, percent=50)

str(core\_tot)

##defining a function "extractCoords" that helps extract coordinates of 50% polygon generated by adehabitatHR

extractCoords <- function(core\_tot)

{

results <- list()

for(i in 1:length(core\_tot@polygons[[1]]@Polygons))

{

results[[i]] <- core\_tot@polygons[[1]]@Polygons[[i]]@coords

}

results <- Reduce(rbind, results)

results

}

#thus, extracting for starters the polygon coods of k50\_total so as to be able to calculate

# the points falling in this polygon

ver\_tot\_polys<-extractCoords(core\_tot)

tot\_pol.x<-ver\_tot\_polys[,1]

tot\_pol.y<-ver\_tot\_polys[,2]

tot\_obs.pts<-point.in.polygon(sub\_dat$X, sub\_dat$Y, pol.x, pol.y, mode.checked=FALSE) #how many pts fall within 50% UD

tot\_obs.pts1<-subset(tot\_obs.pts,tot\_obs.pts=="1")

tot\_obs.pts2<-subset(tot\_obs.pts,tot\_obs.pts=="2")

pts.in\_tot<-sum(tot\_obs.pts1)+length(tot\_obs.pts2)

prop.obs\_tot<-pts.in\_tot/nrow(sub\_dat)

kernel\_tot<-kernelUD(idsp[,1], h = "href", grid = 64, hlim=c(0,3),

same4all = TRUE, kern = c("epa"), extent = 0.5)

kernelUD(idsp[,1])

image(kernel\_tot)

#####kernelUD output- IMG 1 #####

core\_tot <- getverticeshr(kernel\_tot, percent=50)

class(core\_tot)

as.data.frame(core\_tot)

plot(core\_tot, display="filled.contour2", xlab="X", ylab="Y", add=T, lwd=3)

points(dfsp, cex=0.2, pch=16)

points(mean(tot\_pol.x),mean(tot\_pol.y), col="blue", pch="\*", cex= 3)

image(vol\_tot\_ud[[1]],cont=50)

title("Output of getvolumeUD-total for nest v57")

xyzv <- as.image.SpatialGridDataFrame(vol\_tot\_ud[[1]])

contour(xyzv, add=TRUE, percent=50)

points(mean(tot\_pol.x),mean(tot\_pol.y), col="blue", pch="\*", cex= 3)

#########for each animal in this nest

ind.area50<-as.numeric()

ind.area95<-as.numeric()

animalId<-as.numeric()

pts.in<-as.numeric()

prop.obs<-as.numeric()

in.grids<-as.numeric()

tot.locs<-as.numeric()

animalId<-levels(sub\_dat$id)

animalId[8]

summary(sub\_dat)

length(subset(sub\_dat,id=="\_x"))

area50prop<-as.numeric()

ver\_1<-as.numeric()

library(adehabitatHR)

pol.allx<-list()

pol.ally<-list()

mpol.x<-as.numeric()

mpol.y<-as.numeric()

ed\_centre<-as.numeric()

for(j in 1:length(animalId)){

dat1<-subset(sub\_dat, id==animalId[j]) #from my within nest datapts, subset the first animal's (id) datapts

dat1 #data for only 1 animal

x1<-dat1[,4]

y1<-dat1[,5]

tot.locs[j]<-length(dat1[,4]) ##storing the total datapts

xy1<-cbind(x1,y1)

df1<-data.frame(X=xy1[,1],Y=xy1[,2])

dfsp1 <- SpatialPoints(df1)

idsp1<-data.frame(dat1[,2])

sppt1<-data.frame(dfsp1)

coordinates(idsp1)<-sppt1

##for each animal

kernel\_1<-kernelUD(idsp1[,1], h = "href", grid = 64, hlim=c(0,3),

same4all = TRUE, kern = c("epa")

, extent = 1.5

)

par(mfcol=c(1,1))

image(kernel\_1)

core\_1 <- getverticeshr(kernel\_1, percent=50)

ver\_polys<-extractCoords(core\_1)

head(ver\_polys)

head(df1)

point.x<-df1[,1] #so as to extract the x and y coods of the 50% UD

point.y<-df1[,2]

pol.x<-ver\_polys[,1]

pol.y<-ver\_polys[,2]

mpol.x[j]<-mean(pol.x)

mpol.y[j]<-mean(pol.y)

kernel\_1<-kernelUD(idsp1[,1], h = "href", grid = 64, hlim=c(0,3),

same4all = TRUE, kern = c("epa")

, extent = 1.5

)

par(mfcol=c(1,1))

image(kernel\_1)

plot(core\_1, display="filled.contour2", xlab="X", ylab="Y", add=T, lwd=1)

points(dfsp1, cex=0.2, pch=16, add=T)

points(mean(mpol.x[j]),mean(mpol.y[j]), col="blue", pch="\*", cex= 3, add= TRUE)

##labelling the img

library(calibrate)

lab<-textxy(4,6,labs=animalId[j],cex=2)

##plotting nest boundary

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

xynest<-cbind(xnest,ynest)

dfnest<-data.frame(X=xynest[,1],Y=xynest[,2])

head(dfnest)

library(sp)

mapnest<-SpatialPointsDataFrame(coords=dfnest,data=nest)

plot(mapnest, add=TRUE,col="blue", pch="\*")

points(mapnest,lwd=1.2, type= "l")

as.data.frame(core\_1)

ind.area50[j] <- kernel.area(kernel\_1, percent=50)

ind.area95[j] <- kernel.area(kernel\_1, percent=95)

###volume of kernel- for IMG 2 #####

vol\_ud1 <- getvolumeUD(kernel\_1)

image(vol\_ud1[[1]],cont=50)

title("Output of getvolumeUD for nest v57")

xyzv1 <- as.image.SpatialGridDataFrame(vol\_ud1[[1]])

contour(xyzv1, add=TRUE, percent=50)

points(mean(mpol.x[j]),mean(mpol.y[j]), col="blue", pch="\*", cex= 3, add= TRUE)

points(dfsp1, cex=0.2, pch=16)

##labelling the img

library(calibrate)

lab<-textxy(4,7,labs=animalId[j],cex=2)

##plotting nest boundary

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

xynest<-cbind(xnest,ynest)

dfnest<-data.frame(X=xynest[,1],Y=xynest[,2])

head(dfnest)

library(sp)

mapnest<-SpatialPointsDataFrame(coords=dfnest,data=nest)

plot(mapnest, add=TRUE,col="blue", pch="\*")

points(mapnest,lwd=1.2, type= "l")

ver\_polys<-extractCoords(core\_1)

head(ver\_polys)

head(df1)

point.x<-df1[,1] #so as to extract the x and y coods of the 50% UD

point.y<-df1[,2]

pol.x<-ver\_polys[,1]

pol.y<-ver\_polys[,2]

mpol.x[j]<-mean(pol.x)

mpol.y[j]<-mean(pol.y)

ed\_centre[j]<-sqrt((mpol.x[j]-mean(tot\_pol.x))^2 + (mpol.y[j]-mean(tot\_pol.y))^2)

obs.pts<-point.in.polygon(point.x, point.y, pol.x, pol.y, mode.checked=FALSE) #how many pts fall within 50% UD

obs.pts1<-subset(obs.pts,obs.pts=="1")

obs.pts2<-subset(obs.pts,obs.pts=="2")

pts.in[j]<-sum(obs.pts1)+length(obs.pts2)

prop.obs[j]<-pts.in[j]/length(df1[,2]) #of all the pts observed for this animal, how many fall in 50% UD

#####

##plotting nest boundary

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

xynest<-cbind(xnest,ynest)

dfnest<-data.frame(X=xynest[,1],Y=xynest[,2])

head(dfnest)

library(sp)

mapnest<-SpatialPointsDataFrame(coords=dfnest,data=nest)

plot(mapnest, add=TRUE,col="brown", pch=".")

points(mapnest,lwd=3, type= "l", col="brown")

contour(xyzv1, levels=50,add=TRUE, percent=50, col = "green",lwd=3,labcex=1, labels="50%")

pol.allx<-c(pol.allx, list(pol.x))

pol.ally<-c(pol.ally, list(pol.y))

Sys.sleep(1)

}

#nest area#

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

xynest<-cbind(xnest,ynest)

dfnest<-data.frame(X=xynest[,1],Y=xynest[,2])

map\_nest<-SpatialPointsDataFrame(coords=dfnest,data=dfnest)

dummy\_nestid<-rep("v57",times=length(xnest))

length(dummy\_nestid)

dfsp\_nest <- SpatialPoints(dfnest)

idsp\_nest<-data.frame(dummy\_nestid)

sppt\_nest<-data.frame(dfsp\_nest)

coordinates(idsp\_nest)<-sppt\_nest

nest.area<-mcp.area(idsp\_nest[,1], percent=100)

nest.area<-as.numeric(nest.area)

nest.area

#area50 as prop of nest area

obs\_ind.area50<-unlist(ind.area50, recursive = TRUE, use.names = TRUE)

obs\_area50prop<-obs\_ind.area50/nest.area

## my test statistic observed ##

obs\_ratio\_ppap<-prop.obs/obs\_area50prop

obs\_ratio\_ppap

obs\_test.ratio<-cbind(animalId, obs\_ratio\_ppap)

obs\_test.ratio

#OVERLAP#

idsp\_ovrlp<-data.frame(sub\_dat[,1])

coordinates(idsp\_ovrlp)<-dftot

##for the entire nest

kernel\_ovr<-kernelUD(idsp\_ovrlp[,1], h = "href", grid = 64, hlim=c(0,3),

same4all = TRUE, kern = c("epa"),

#extent = 0.5

)

overlap\_all<- kerneloverlaphr(kernel\_ovr,method="HR",lev=50)

View(overlap\_all)

#GENERAL#

xycoods<-SpatialPoints(cbind(sub\_dat$X,sub\_dat$Y))

head(xycoods)

xy\_unique<-remove.duplicates(xycoods)

xy\_unique<-as.data.frame(xy\_unique)

#length(sub\_img$X) ##only for nest v57

length(dat$X)

length(sub\_dat$X)

length(xy\_unique[,1])

#NEST SUMMARY#

nest\_summary<-cbind(levels(sub\_dat$id), prop.obs, obs\_ind.area50,ind.area95, obs\_area50prop)

View(nest\_summary)

length(animalId)

length(sub\_dat$id=="\_X")

################

kernel\_all<-kernelUD(df,id = sub\_dat$id, h = "href", grid = 64,

hlim= c(0.1,1.5),same4all = TRUE, kern = "epa", extent = 0.5)

overlap\_all<- kerneloverlaphr(kernel\_all,method="HR",lev=50)

1. **Spatial fidelity**

iter<-1000

library(sp)

xycoods<-SpatialPoints(cbind(sub\_dat$X,sub\_dat$Y))

head(xycoods)

xy\_unique<-remove.duplicates(xycoods) #remove duplicated xy datapts

head(xy\_unique)

class(xy\_unique)

#sample(xy\_unique,1)

xy\_unique<-as.data.frame(xy\_unique)

z<-xy\_unique[sample(nrow(xy\_unique), 1), ]

z

animalId

length(xycoods)

#sample(xy\_unique,1)

nrow(xy\_unique) ##removed a few repeated coods =51

#xcood<-as.data.frame(xy\_unique$`coords.x1`)

xpick<-as.numeric()

ypick<-as.numeric()

draw.pt<-as.numeric() #empty vector for storing random x coods

#drawY<-as.numeric() #empty vector for storing random y coods

avg<-as.numeric() #empty vector to store average area over 1000 iters for each animal

std<-as.numeric() #empty vector to store average area over 1000 iters for each animal

iter\_area50<-as.numeric() #empty vector to store absolute area from each of the iterations

iter\_area50prop<-as.numeric()

iter\_test.ratio<-as.numeric()

iter\_pts.in<-as.numeric()

iter\_prop.obs<-as.numeric()

for(k in 1:length(animalId)){ #for each animal

for(j in 1:iter){ #for each iteration

draw.pt<-0 #before picking random x and y, set the counter to 0

#for(i in 1:tot.locs[k]){ # pick x and y coods randomly as many times (tot.locs) as that

draw.pt<-xy\_unique[sample(nrow(xy\_unique), tot.locs[k],replace=TRUE), ] # particular animal was actually seen in the observations made (eg. out of a total of 100 scans drawY[i]<-sample(yoptions,1) # if the animal was seen only 25 times, then pick x and y 25 times)

#sample the non-repeated data points

xpick<-as.data.frame(draw.pt$`coords.x1`) ##random x coods sampled as many times as the animal was seen

ypick<-as.data.frame(draw.pt$`coords.x2`) ##random y coods sampled as many times as the animal was seen

dfpick<-data.frame(X=xpick,Y=ypick) ##make a dataframe

library(sp)

dummy\_id<-rep("v57",times=tot.locs[k])

iter\_dfsp <- SpatialPoints(dfpick)

iter\_idsp<-data.frame(dummy\_id)

iter\_sppt<-data.frame(iter\_dfsp)

coordinates(iter\_idsp)<-iter\_sppt

##for each iteration

kernel\_iter<-kernelUD(iter\_idsp[,1], h = "href", grid = 64, hlim=c(0,3),

same4all = TRUE, kern = c("epa"), extent = 2)

core\_iter <- getverticeshr(kernel\_iter, percent=50)

as.data.frame(core\_iter)

##calculating area for the randomly sampled coordinates in each iteration

iter\_area50[j] <- kernel.area(kernel\_iter, percent=50)

##\*\*\*\*calculating proportion of area50 by nest area

class(iter\_area50)

iter\_area50<-unlist(iter\_area50, recursive = TRUE, use.names = TRUE)

#nest.area<-rep(nest.area,times=iter)

iter\_area50prop<-iter\_area50/nest.area #nest.area from kernel code page

iter\_ver\_polys<-extractCoords(core\_iter)

head(iter\_ver\_polys)

head(dfpick)

iter\_point.x<-dfpick[,1] #so as to extract the x and y coods of the picked at random points

iter\_point.y<-dfpick[,2]

iter\_pol.x<-iter\_ver\_polys[,1]

iter\_pol.y<-iter\_ver\_polys[,2]

iter\_obs.pts<-point.in.polygon(iter\_point.x, iter\_point.y, iter\_pol.x, iter\_pol.y, mode.checked=FALSE) #how many pts fall within 50% UD

iter\_obs.pts1<-subset(iter\_obs.pts,iter\_obs.pts=="1")

iter\_obs.pts3<-subset(iter\_obs.pts,iter\_obs.pts=="3")

iter\_pts.in<-length(iter\_obs.pts1)+length(iter\_obs.pts3)

iter\_prop.obs[j]<-iter\_pts.in/nrow(dfpick) #of all the pts observed for this animal, how many fall in 50% UD

## for test statistic : ratio of prop. obs and area 50% prop of nest

#length(area50prop)

iter\_test.ratio<-iter\_prop.obs/iter\_area50prop

iter\_test.ratio<-as.numeric(iter\_test.ratio)

}

avg[k]<-mean(iter\_test.ratio) ##the average area across 1000 iterations, for each animal

std[k]<-sd(iter\_test.ratio) ##the std. dev. across 1000 iterations, for each animal

#limits[k]<-c(avg+2\*std,avg-2\*std)

hist(iter\_test.ratio, xlim= c(0,14), main="ratio: proportion of data points in core area as a proportion of nest area") #plotting the area for each animal

abline(v=avg[k]+2\*std[k],lty=2)

abline(v=avg[k]-2\*std[k],lty=2) ##the confidence interval for p<0.05

abline(v=obs\_ratio\_ppap[k], col="red",lwd=3) ##the actual area calculated previously for each animal

abline(v=avg[k], lty="dotted",lwd=3, col="green") #plotting the mean across iters

library(calibrate)

lab<-textxy(4,50,labs=animalId[k],cex=2) ##adding a label for each animal to the plot

Sys.sleep(1) ##viewing each plot at a delay of 1 sec

}

1. **Dyadic interactions**

setwd("E:/")

dat<-read.csv("overlap\_allnests.csv")

dat<-subset(dat,dat$overlap\_Y<1) ##remove overlaps with self

dat[is.na(dat)]<-0

nrow(dat)

head(dat)

library(glmmTMB)

mod1<-glmmTMB(cbind(overlap\_Y,overlap\_N) ~ DB + SC + ST + (1 | nest),

data=dat,

family=betabinomial(link = "logit"),

na.action = na.omit)

summary(mod1)

confint(mod1)

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), DB=numeric(iter),

SC=numeric(iter), ST=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest) %>% arrange(nest)

table(mftable.new$nest)

for (i in 1:iter){

null.dat<- mftable.new %>% dplyr::select(SC,DB,ST,overlap\_Y,overlap\_N,nest,from,to) %>% sample\_frac(size=1)

null.dat$DB <- mftable.new$DB

null.dat$SC <- mftable.new$SC

null.dat$ST <- mftable.new$ST

#store[i,] <- summary(lmer((overlap\_Y) ~ DB+SC+ST+(1|nest), data=null.dat))$coefficients[1:4]

store[i,] <- summary(glmmTMB(cbind(overlap\_Y,overlap\_N) ~ DB + SC + ST + (1 | nest),

data=null.dat,

family=betabinomial(link = "logit"),

na.action = na.omit

))$coefficients$cond[1:4]

print(i)

}

coef.mod <- summary(mod1)$coefficients$cond[1:4]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), DB=numeric(iter),

SC=numeric(iter), ST=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest = with(dat, sample(unique(nest), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

print(i)

store[i,] <- summary(glmmTMB(cbind(overlap\_Y,overlap\_N) ~ DB + SC + ST + (1 | newcat),

data=boot.d,

family=betabinomial(link = "logit"),

na.action = na.omit))$coefficients$cond[1:4]

}

store[1:4,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

m<-summary(glmmTMB(cbind(overlap\_Y,overlap\_N) ~ DB + SC + ST + (1 | newcat),

data=boot.d,

family=betabinomial(link = "logit"),

na.action = na.omit))$coefficients$cond[1:4]

cbind(estimate=m, t(out))

1. **Predictors of spatial fidelity**

dat<-read.csv("**raw data**.csv")

str(all)

max(all$DB)

library(lme4)

dat$Significance<-as.factor(as.character(dat$Significance))

mod<-glmer(Significance ~ FL+ MA+ Sub+DB+ FG

+(1|nest),family=binomial, data=dat) #&Sub &DB

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest) %>% arrange(nest)

table(mftable.new$nest)

for (i in 1:iter){

null.dat <- mftable.new %>% select(Significance,FL,MA,Sub,DB,FG,nest) %>% sample\_frac(size=1) ## change

null.dat$FL <- mftable.new$FL

null.dat$MA <- mftable.new$MA

null.dat$Sub <- mftable.new$Sub

null.dat$DB <- mftable.new$DB

null.dat$FG <- mftable.new$FG

store[i,] <-summary(glmer(Significance ~ FL+ MA+ Sub+DB+ FG+(1|nest),family=binomial, data=null.dat))$coefficients[1:6]

print(i)

}

##

coef.mod <- summary(mod)$coefficients[1:6]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest = with(dat, sample(unique(nest), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

print(i)

store[i,] <-summary(glmer(Significance ~ FL+ MA+ Sub+DB+ FG+(1|newcat),family=binomial, data=boot.d))$coefficients[1:6]

}

store[1:6,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

mod<-summary(glmer(Significance ~ FL+ MA+ Sub+DB+ FG+(1|nest),family=binomial, data=dat))$coefficients[1:6]

cbind(estimate=mod, t(out))

mm\_sig.fidelity<-glmer(Significance ~ FL+ MA+ Sub+DB+ FG+(1|nest),family=binomial, data=dat)

summary(mm\_sig.fidelity)

1. **Predictors of core area size**

all<-read.csv("**raw data**.csv")

dat<-all[which(all$Significance == "1"),] ##ONLY THE ONES THAT SHOW SPATIAL FIDELITY

nrow(all)

library(lme4)

mod<-lmer(UD50byarea ~ FL+MA+Sub+DB+FG+(1|nest) , data=dat)

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest) %>% arrange(nest)

table(mftable.new$nest)

for (i in 1:iter){

null.dat <- mftable.new %>% select(UD50byarea,FL,MA,Sub,DB,FG,nest) %>% sample\_frac(size=1) ## change

null.dat$FL <- mftable.new$FL

null.dat$MA <- mftable.new$MA

null.dat$Sub <- mftable.new$Sub

null.dat$DB <- mftable.new$DB

null.dat$FG <- mftable.new$FG

store[i,] <-summary(lmer(UD50byarea ~ FL+MA+Sub+DB+FG+(1|nest) , data=null.dat))$coefficients[1:6]

print(i)

}

##

coef.mod <- summary(mod)$coefficients[1:6]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest = with(dat, sample(unique(nest), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

print(i)

store[i,] <-summary(lmer(UD50byarea ~ FL+MA+Sub+DB+FG+(1|newcat) , data=boot.d))$coefficients[1:6]

}

store[1:6,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

mod<-summary(lmer(UD50byarea ~ FL+MA+Sub+DB+FG+(1|nest) , data=dat))$coefficients[1:6]

cbind(estimate=mod, t(out))

mm\_area<-(lmer(UD50byarea ~ FL+MA+Sub+DB+FG+(1|nest) , data=dat))

1. **Predictors of location of a wasp**

ed\_centre vs. beh

dat<-read.csv("**raw data**.csv")

library(lme4)

mod<-lmer(ed\_centre ~ FL+MA+Sub+DB+FG+(1|nest) , data=dat)

summary(mod)

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest) %>% arrange(nest)

table(mftable.new$nest)

str(mftable.new)

for (i in 1:iter){

null.dat <- mftable.new %>% dplyr::select(ed\_centre,FL,MA,Sub,DB,FG,nest) %>% sample\_frac(size=1) null.dat$FL <- mftable.new$FL

null.dat$MA <- mftable.new$MA

null.dat$Sub <- mftable.new$Sub

null.dat$DB <- mftable.new$DB

null.dat$FG <- mftable.new$FG

store[i,] <-summary(lmer(ed\_centre ~ FL+MA+Sub+DB+FG+(1|nest) , data=null.dat))$coefficients[1:6]

print(i)

}

##

coef.mod <- summary(mod)$coefficients[1:6]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest = with(dat, sample(unique(nest), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

print(i)

store[i,] <-summary(lmer(ed\_centre ~ FL+MA+Sub+DB+FG+(1|newcat) , data=boot.d))$coefficients[1:6]

}

store[1:6,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

mod<-summary(lmer(ed\_centre ~ FL+MA+Sub+DB+FG+(1|nest) , data=dat))$coefficients[1:6]

cbind(estimate=mod, t(out))

mm\_area<-(lmer(ed\_centre ~ FL+MA+Sub+DB+FG+(1|nest) , data=dat))

summary(mm\_area)

1. **Activity = step length - for each nest separately**

#if the slice is consecutive, calculate distance from previous slice

all<-read.csv("**v57\_day123\_raw**.csv")

##to remove all off nest points

tail(all)

dat<- all[!(is.na(all$X)),]

length(levels(dat$id))

levels(dat$id)

length(dat[,4])

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

#head(dat)

datx<-dat[,4]

daty<-dat[,5]

pol.x<-xnest

pol.y<-ynest

library(sp)

overall.pts<-point.in.polygon(datx,daty, pol.x, pol.y, mode.checked=FALSE)

dataset<-dat[,c(1,2,3,4,5,6)]

head(dataset)

include.pts<-cbind(dataset,overall.pts)

include.pts<-as.data.frame(include.pts)

pt.ins1<-include.pts[ which(overall.pts=='1'),] #pts that fall inside the polygon

pt.ins3<-include.pts[ which(overall.pts=='3'),] #those that fall on the nest boundary

#head(pt.ins3)

dat<-rbind(pt.ins1,pt.ins3)

dat<-rbind(pt.ins1,pt.ins3)

centroidX<-mean(dat$X)

centroidY<-mean(dat$Y)

sub\_Q<-subset(dat,id=="Q")

nrow(sub\_Q) ##each day, there should be at least 100 frames

animalId<-levels(dat$id)

day\_ed<-list()

ind\_ed<-list()

x\_sq\_diff<-list()

avg\_1day<-as.numeric()

ed<-list()

ed\_avgof3<-list()

id\_dist<-list()

Q.w\_ed<-as.numeric()

Q.w\_med\_ed<-as.numeric()

day\_ed<-as.numeric

daywise\_each\_ed<-list()

instantaneous\_ind\_Q\_ed<-list()

frames\_day<-list()

frames\_id<-list()

consecutive\_d1<-as.numeric()

length(consecutive\_d1)

#ind\_dat$Slice

cons\_eds<-list()

tot.cons\_eds1<-list()

non\_cons\_eds<-list()

tot.cons\_eds2<-list()

tot.ed1\_id<-list()

tot.ed2\_id<-list()

ED\_2frames<-list()

##pick an animal

tot.ed1\_id<-list()

tot.cons\_eds1<-list()

tot.non.cons\_eds2<-list()

total.day.id<-list()

velocity.cons.id<-list()

avg\_vel\_id<-as.numeric()

vel\_all.ids<-list()

id\_step.length<-as.numeric()

for(j in 1:length(animalId)){

ind\_dat<-subset(dat, id==animalId[j]) ##subset for that animal

ind\_dat$day<-as.factor(ind\_dat$day)

non.cons\_frames<-as.numeric()

cons\_frames<-as.numeric()

total\_day<-as.numeric()

velocity.cons<-as.numeric()

step.length<-as.numeric()

##the no. of days that animal was observed

for(i in levels(ind\_dat$day[j])){

ind\_day<-subset(ind\_dat, day==i) ##subset for that day

consecutive\_d1<-as.numeric() ##empty the counter before each

cons\_eds<-list()

for(k in 1:(length(ind\_day$Slice)-1)){

##because you don't want to go beyond second last+1=last frame

if (ind\_day$Slice[k]==ind\_day$Slice[k+1]-1){ ##check if the frames are consecutive

consecutive\_d1[k]<-(sqrt((ind\_day$X[k]-ind\_day$X[k+1])^2 +(ind\_day$Y[k]-ind\_day$Y[k+1])^2)) ##store each consecutive distance of a wasp

##calculate ED if the frames are consecutive

}

ED\_2frames<-cbind(ind\_day$Slice[k],consecutive\_d1[k])

cons\_eds[k]<-c(list(ED\_2frames))

}

if(length(which(consecutive\_d1 != "NA"))>10){

non.cons\_frames[i]<-99-length(which(consecutive\_d1 != "NA"))

##for each day, how many consecutive frame distances?

#only considering 100 because that wasp was alive on that nest that day

step.length[i]<-mean(na.omit(as.vector(consecutive\_d1)))

tot.cons\_eds1[i]<-sum(na.omit(as.vector(consecutive\_d1)))##pick EDs that are not NA ##total distance travelled each day for consecutive frames, excluding NAs- non-cons frames

tot.non.cons\_eds2<-5.6\*non.cons\_frames ##no [i] because non.cons\_frames is already a vector with [i]

##total distance travelled each day for non-consecutive frames

total\_day[i]<-(as.numeric(tot.cons\_eds1[i])+as.numeric(tot.non.cons\_eds2[i]))

}

}

id\_step.length[j]<-mean(step.length)

tot.ed1\_id[j]<-c(list(tot.cons\_eds1)) ##for each animal

tot.ed2\_id[j]<-c(list(tot.non.cons\_eds2))

total.day.id[j]<-c(list(total\_day))

}

1. **Role of brood- for each nest**

all\_nests<-read.csv("**brood\_allnests.csv**")

brood<-subset(all\_nests, nest=="v57")

library(adehabitatHR)

xbrood<-brood$X

ybrood<-brood$Y

xybrood<-cbind(xbrood,ybrood, add=T)

plot(xbrood,ybrood)

dfbrood<-data.frame(X=xybrood[,1],Y=xybrood[,2])

dfspbrood <- SpatialPoints(xybrood)

#empty locs

emp<-brood[which(brood$contents=="X"),]

xemp<-emp$X

yemp<-emp$Y

xyemp<-cbind(xemp,yemp)

dfemp<-data.frame(X=xyemp[,1],Y=xyemp[,2])

dfspemp <- SpatialPoints(xyemp)

plot(dfspemp,pch=4, add=T)

#egg locations

e<-brood[which(brood$contents=="E"),]

xegg<-e$X

yegg<-e$Y

xyegg<-cbind(xegg,yegg)

dfegg<-data.frame(X=xyegg[,1],Y=xyegg[,2])

dfspegg <- SpatialPoints(xyegg)

plot(dfspegg,pch=1, add=T)

## LARVAE locs

L<-brood[which(brood$contents=="L"),]

xL<-L$X

yL<-L$Y

xyL<-cbind(xL,yL)

dfL<-data.frame(X=xyL[,1],Y=xyL[,2])

dfspL <- SpatialPoints(xyL)

plot(dfspL, add=T, pch="L")

#### pupae locs

p<-brood[which(brood$contents=="P"),]

xp<-p$X

yp<-p$Y

xyp<-cbind(xp,yp)

dfp<-data.frame(X=xyp[,1],Y=xyp[,2])

p$no<-as.character(c(1:nrow(p)))

dfspp <- SpatialPoints(xyp)

plot(dfspp, add=T, pch="P")

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

# plotting random circles with same area on nest #

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

rand.pts<-as.numeric()

rand.pt<-as.numeric()

lar.in.123<-as.numeric()

prop.lar.123<-as.numeric()

rand.buffs<-list()

anim\_3buffs<-list()

anim\_lar123<-list()

anim\_prop123<-list()

radii<-list()

r[29]

for(j in 1:length(animalId)){

#r\_id<-r[j]

##plotting nest boundary

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

xynest<-cbind(xnest,ynest)

dfnest<-data.frame(X=xynest[,1],Y=xynest[,2])

head(dfnest)

library(sp)

mapnest<-SpatialPointsDataFrame(coords=dfnest,data=nest)

points(mean(mapnest$X), mean(mapnest$Y), col="red")

plot(mapnest, col="blue", pch="\*")

points(mapnest,lwd=1.2, type= "l")

########################################

for(i in 1:5){

## draw a random core area five times

##considering it doesn't fall outside the nest boundary

rand.pt<-sample(xy\_unique,1) #sample 1 random pt around which a buffer will be made)

#pol.allx<-c(pol.allx, list(pol.x))

#rand.pts<-c(rand.pt,list(rand.pt))

class(rand.pt) ##has to be sp

plot(dfspbrood, add=T)

plot(rand.pt, add=T, col="red")

library(rgeos)

class(r)

while (apply(gDistance((rand.pt), mapnest,byid=TRUE),2,min) <r[j]\*100) { ##while helps resample untill condition is met

##if the distance of the sampled random point from nest boundary is greater

##than the radius of buffer to be made, proceed plotting and calculating larvae inside

rand.pt<-sample(xy\_unique,1) #sample 1 random pt around which a buffer will be made)

class(rand.pt) ##has to be sp

plot(rand.pt, add=T, col="red")

}

rand.pts[i]<-c(rand.pt)

class(rand.pts)

library(rgeos)

rand.buff<-(gBuffer(SpatialPoints(rand.pts[i]), width=r[j]\*100, byid=TRUE)) ##first wasp's radius based on observed area50%

lar.of.all<-over(dfspL,rand.buff)

lar.in<-length(which(lar.of.all>0))

rand.buffs[i]<-c(rand.buff)

lar.in.123[i]<-c(lar.in)

prop.lar.123[i]<-lar.in.123[[i]]/length(lar.of.all)

class(rand.buffs)

#rand.buffs<-unlist(rand.buffs)

plot(rand.buffs[[i]], add=T)

##around the first random pt, draw a circle of that animal's

#earlier calculated radius

} ##select another random point and make a buffer around that too

Sys.sleep(1)

#pol.allx<-c(pol.allx, list(pol.x))

anim\_3buffs[j]<-c(rand.buffs[[i]])

anim\_lar123[j]<-list(lar.in.123)

anim\_prop123[j]<-list(prop.lar.123)

# radii[j]<-list(r\_id)

length(radii)

}

animalId<-as.vector(animalId)

rand.lar <- sapply(anim\_lar123, paste0, collapse=",") ## attaching a list to dataframe

rand.lar.prop<-sapply(anim\_prop123, paste0, collapse=",")

random.core<-(cbind(animalId,rand.lar,rand.lar.prop))

all<-read.csv("**compos\_linear regression**.csv")

tail((all))

dat<-subset(all, Significance=="1")

dat0<-subset(all, Significance=="0")##those that do not show fidelity at all

summary(all)

nrow(all)

library(mclogit)

library(coxme)

#all$occupancy<-as.factor(all$occupancy)

mod1<-coxme(Surv(dummy,occupancy)~rand.lar.prop+strata(unique\_id)+

(rand.lar.prop|nest), data=dat)

summary(mod1)

##Mixed Conditional logistic regression

library(TwoStepCLogit)

Ts.estim(formula = occupancy ~ rand.lar.prop+ strata(unique\_id)+ cluster(nest), #Used is the column with your 0s and 1s

data = all, #Your data

random = ~ rand.lar.prop)

1. **Real time distance of worker from queen – separately for each nest**

dat<-read.csv("**v57\_day123\_raw**.csv")

sub\_Q<-subset(dat,id=="Q")

str(dat)

animalId<-levels(dat$id)

day\_ed<-list()

ind\_ed<-list()

x\_sq\_diff<-list()

avg\_1day<-as.numeric()

ed<-list()

ed\_avgof3<-list()

id\_dist<-list()

for(j in 1:length(animalId)){

ind\_dat<-subset(dat, id==animalId[j])

ind\_dat$day<-as.factor(ind\_dat$day)

for(i in 1:length(levels(ind\_dat$day))){

ind\_day<-subset(ind\_dat, day==i)

ind\_day<-ind\_day[!duplicated(ind\_day$Slice),]

Q\_day<-subset(sub\_Q, day==i)

m<-match(ind\_day$Slice,Q\_day$Slice, NA, incomparables = NULL)

ind\_day<-cbind(ind\_day,m)

ind\_day<- ind\_day[!(is.na(ind\_day$m)),] ##firstly remove the slices in ind that are absent from queen's slice

mQ<-match(Q\_day$Slice,ind\_day$Slice, NA, incomparables = NULL)

Q\_day<-cbind(Q\_day,mQ)

Q\_day<- Q\_day[!(is.na(Q\_day$mQ)),]

all<-cbind(Q\_day,ind\_day)

nrow(Q\_day)

nrow(ind\_day)

x\_sq\_diff<-(ind\_day$X-Q\_day$X)

day\_ed<-sqrt((ind\_day$X-Q\_day$X)^2 + (ind\_day$Y-Q\_day$Y)^2)

avg\_1day[i]<-mean(day\_ed)

}

ed[j]<- list(avg\_1day)

ed\_avgof3[j]<-mean(avg\_1day)

avg\_ed<-unlist(ed\_avgof3)

euc\_dist<-cbind(animalId,avg\_ed,ed)

}

View(euc\_dist)

**## Real-time distance from queen and behaviour**

all<-read.csv("**raw data**.csv", na.strings = c("NA","","NaN"))

dat<-subset(all, id !="Q") ##removed the queen because overlap with self does not make sense

##ED does not consider core areas anyhow

library(lme4)

mod<-lmer((avg\_ed) ~ FL+MA+Sub+DB+FG+ (1|nest), data=dat)

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest) %>% arrange(nest)

table(mftable.new$nest)

for (i in 1:iter){

null.dat <- mftable.new %>% select(avg\_ed,FL,MA,Sub,DB,FG,nest) %>% sample\_frac(size=1) ## change

null.dat$FL <- mftable.new$FL

null.dat$MA <- mftable.new$MA

null.dat$Sub <- mftable.new$Sub

null.dat$DB <- mftable.new$DB

null.dat$FG <- mftable.new$FG

store[i,] <-summary(lmer((avg\_ed) ~ FL+MA+Sub+DB+FG+ (1|nest),data=null.dat))$coefficients[1:6]

print(i)

}

coef.mod <- summary(mod)$coefficients[1:6]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), FL=numeric(iter),MA=numeric(iter),Sub=numeric(iter),

DB=numeric(iter), FG=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest = with(dat, sample(unique(nest), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

print(i)

store[i,] <-summary(lmer((avg\_ed) ~ FL+MA+Sub+DB+FG+(1|newcat),data=boot.d))$coefficients[1:6]

}

nrow(dat)

store[1:6,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

mod<-summary(lmer((avg\_ed) ~ FL+MA+Sub+DB+FG+(1|nest),data=dat))$coefficients[1:6]

cbind(estimate=mod, t(out))

**# off-nest points**

dat<-read.csv("**v57\_day123\_raw**.csv")

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

animalId<-levels(dat$id)

class(animalId)

in.nest<-as.numeric()

off.nest<-as.numeric()

for (i in 1:length(animalId)){

dat1<-subset(dat, id==animalId[i])

nrow(dat1)

dat1x<-dat1[,4]

dat1y<-dat1[,5]

pol.x<-xnest

pol.y<-ynest

library(sp)

overall.pts<-point.in.polygon(dat1x,dat1y, pol.x, pol.y, mode.checked=FALSE)##how many points fall inside nest

off.nest[i]<-length(which(overall.pts=='0'))

dataset<-dat1[,c(1,2,3,4,5)]

head(dataset)

nrow(dataset)

include.pts<-cbind(dataset,overall.pts)

include.pts<-as.data.frame(include.pts)

pt.ins1<-include.pts[ which(overall.pts=='1'),] #pts that fall inside the polygon

pt.ins3<-include.pts[ which(overall.pts=='3'),] #those that fall on the nest boundary

#head(pt.ins3)

sub\_dat<-rbind(pt.ins1,pt.ins3) #row bind for all x,y that are within nest polygon

in.nest[i]<-nrow(sub\_dat)

plot(dat1$X,dat1$Y, main=animalId[i],xlab="x cood", ylab= "y cood", xlim=c(0,16), ylim=c(0,16))

##plotting nest boundary

nest<-read.csv("nest\_v57\_1.csv")

xnest<-nest[,1]

ynest<-nest[,2]

xynest<-cbind(xnest,ynest)

dfnest<-data.frame(X=xynest[,1],Y=xynest[,2])

head(dfnest)

library(sp)

mapnest<-SpatialPointsDataFrame(coords=dfnest,data=nest)

plot(mapnest, add=TRUE,col="blue", pch="\*")

points(mapnest,lwd=1.2, type= "l")

Sys.sleep(1)

}

percent.off<-(off.nest/(off.nest+in.nest))\*100

View(cbind(animalId,off.nest, in.nest, percent.off))

1. **Ovarian index**

dist\_Q<-read.csv("**dist\_Q.csv**")

ovary<-read.csv("**ovarian index2**.csv")

data<-read.csv("**alldata\_activity**.csv")

all<-data[!is.na(data$step.length),]

dist\_v14<-subset(dist\_Q, nest=="v14")

ov\_v14<-subset(ovary, nest=="v14")

all\_v14<-subset(all, nest=="v14")

v14<-merge(dist\_v14,ov\_v14, by="id")

dat\_v14<-merge(v14,all\_v14,by="id")

#names(v14\_max.ed)[names(v14\_max.ed) == 'cell\_ed.x'] <- 'cell\_ed'

dist\_v87<-subset(dist\_Q, nest=="v87")

ov\_v87<-subset(ovary, nest=="v87")

v87<-merge(dist\_v87,ov\_v87, by="id")

all\_v87<-subset(all, nest=="v87")

dat\_v87<-merge(v87,all\_v87,by="id")

dist\_v72<-subset(dist\_Q, nest=="v72")

ov\_v72<-subset(ovary, nest=="v72")

v72<-merge(dist\_v72,ov\_v72, by="id")

all\_v72<-subset(all, nest=="v72")

dat\_v72<-merge(v72,all\_v72,by="id")

dist\_v82<-subset(dist\_Q, nest=="v82")

ov\_v82<-subset(ovary, nest=="v82")

v82<-merge(dist\_v82,ov\_v82, by="id")

all\_v82<-subset(all, nest=="v82")

dat\_v82<-merge(v82,all\_v82,by="id")

dist\_v99<-subset(dist\_Q, nest=="v99")

ov\_v99<-subset(ovary, nest=="v99")

v99<-merge(dist\_v99,ov\_v99, by="id")

all\_v99<-subset(all, nest=="v99")

dat\_v99<-merge(v99,all\_v99,by="id")

all\_oi\_ed<-rbind(dat\_v72,dat\_v82,dat\_v87,dat\_v14,dat\_v99)

nrow(all\_oi\_ed)

dat\_withQ<-all\_oi\_ed

boxplot(OI~Q.status, data=all\_oi\_ed, names=c("Worker","Queen","PQ"), ylab="Ovarian index")

all\_oi\_ed<-subset(all\_oi\_ed, id !="Q") ##remove self overlaps of queen

hist(all\_oi\_ed$avg\_ed.x)

library(lme4)

dat\_withQ<-all\_oi\_ed[!is.na(all\_oi\_ed$OI),]

#dat\_1<-subset(dat, Significance=="1")

dat\_noQ<-subset(dat\_withQ, id!="Q")

## lmer : OI vs. ed\_centre ##

#RUN OI vs. ED

dat<-dat\_noQ

nrow(dat)

library(lme4)

mod<-lmer(OI~avg\_ed.y+ (1|nest.x), data=dat)

summary(mod)

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), step.length=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest.x) %>% arrange(nest.x)

table(mftable.new$nest.x)

for (i in 1:iter){

null.dat <- mftable.new %>% select(OI,avg\_ed.y,nest.x) %>% sample\_frac(size=1) ## change

null.dat$avg\_ed.y <- mftable.new$avg\_ed.y

##

store[i,] <-summary(lmer(OI~avg\_ed.y+ (1|nest.x), data=null.dat))$coefficients[1:2]

print(i)

}

##

coef.mod <- summary(mod)$coefficients[1:2]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), avg\_ed.y=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest = with(dat, sample(unique(nest.x), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest.x)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

print(i)

store[i,] <-summary(lmer(OI~avg\_ed.y+ (1|newcat),data=boot.d))$coefficients[1:2]

}

store[1:2,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

mod<-summary(lmer(OI~avg\_ed.y+ (1|nest.x),data=dat))$coefficients[1:2]

cbind(estimate=mod, t(out))

1. **Social Partners**

dat<-read.csv("**raw data**.csv")

setwd("E:/chapter1\_data/six nests/overlaps/clean data/networks/degrees\_all")

ST<-read.csv("**deg.all\_ST**.csv")

DB<-read.csv("**deg.all\_DB**.csv")

SC<-read.csv("**deg.all\_SC**.csv")

dat\_v57<-subset(dat, nest=="v57")

ST\_v57<-subset(ST, nest=="v57")

v57<-merge(dat\_v57,ST\_v57, by="id",all.x="TRUE")

SC\_v57<-subset(SC, nest=="v57")

v57<-merge(v57,SC\_v57, by="id",all.x="TRUE")

DB\_v57<-subset(DB, nest=="v57")

v57<-merge(v57,DB\_v57, by="id",all.x="TRUE")

dat\_v14<-subset(dat, nest=="v14")

ST\_v14<-subset(ST, nest=="v14")

v14<-merge(dat\_v14,ST\_v14, by="id",all.x="TRUE")

SC\_v14<-subset(SC, nest=="v14")

v14<-merge(v14,SC\_v14, by="id",all.x="TRUE")

DB\_v14<-subset(DB, nest=="v14")

v14<-merge(v14,DB\_v14, by="id",all.x="TRUE")

#names(v14\_max.ed)[names(v14\_max.ed) == 'cell\_ed.x'] <- 'cell\_ed'

dat\_v87<-subset(dat, nest=="v87")

ST\_v87<-subset(ST, nest=="v87")

v87<-merge(dat\_v87,ST\_v87, by="id",all.x="TRUE")

SC\_v87<-subset(SC, nest=="v87")

v87<-merge(v87,SC\_v87, by="id",all.x="TRUE")

DB\_v87<-subset(DB, nest=="v87")

v87<-merge(v87,DB\_v87, by="id",all.x="TRUE")

dat\_v72<-subset(dat, nest=="v72")

ST\_v72<-subset(ST, nest=="v72")

v72<-merge(dat\_v72,ST\_v72, by="id",all.x="TRUE")

SC\_v72<-subset(SC, nest=="v72")

v72<-merge(v72,SC\_v72, by="id",all.x="TRUE")

DB\_v72<-subset(DB, nest=="v72")

v72<-merge(v72,DB\_v72, by="id",all.x="TRUE")

dat\_v82<-subset(dat, nest=="v82")

ST\_v82<-subset(ST, nest=="v82")

v82<-merge(dat\_v82,ST\_v82, by="id",all.x="TRUE")

SC\_v82<-subset(SC, nest=="v82")

v82<-merge(v82,SC\_v82, by="id",all.x="TRUE")

DB\_v82<-subset(DB, nest=="v82")

v82<-merge(v82,DB\_v82, by="id",all.x="TRUE")

dat\_v99<-subset(dat, nest=="v99")

ST\_v99<-subset(ST, nest=="v99")

v99<-merge(dat\_v99,ST\_v99, by="id",all.x="TRUE")

SC\_v99<-subset(SC, nest=="v99")

v99<-merge(v99,SC\_v99, by="id",all.x="TRUE")

DB\_v99<-subset(DB, nest=="v99")

v99<-merge(v99,DB\_v99, by="id",all.x="TRUE")

all\_dat<-rbind(v57,v82,v87,v72,v99,v14)

all\_dat[is.na(all\_dat)]<-0 ##replace NAs with 0 ##a$x[is.na(a$x)] <- 0 for column

all\_dat$DBSCST<-(all\_dat$DB\_deg.all+all\_dat$SC\_deg.all+all\_dat$ST\_deg.all)

nrow(all\_dat)

dat<-subset(all\_dat, nest.x!="v14")

nrow(dat)

dat1<-subset(dat, Significance=="1")

nrow(dat1)

all\_dat$avg.deg<-all\_dat$DBSCST/3

all\_int<-subset(all\_dat, avg.deg>0)

all\_int1<-subset(all\_dat, Significance=="1")

mod\_int<-lmer(avg.deg~UD50byarea +(1|nest.x), data=all\_int1)

##The size of core area irrespective of the location is positively correlated with

## number of interactions with the nestmates.

summary(mod\_int)

confint(mod\_int)

m1<-update(mod\_int,~.-UD50byarea)

anova(mod\_int,m1)

################## social partners vs. UD50byarea############

# RUN degree vs. UD50 first

dat<-all\_int1

## avg.deg vs. UD50byarea

library(lme4)

mod<-lmer(avg.deg ~UD50byarea+(1|nest.x),data=dat)

#Randomisation

iter=1000

store <- data.frame(intercept=numeric(iter), UD50byarea=numeric(iter))

library(dplyr)

mftable.new <- dat %>% group\_by(nest.x) %>% arrange(nest.x)

table(mftable.new$nest.x)

for (i in 1:iter){

null.dat <- mftable.new %>% select(avg.deg,UD50byarea,nest.x) %>% sample\_frac(size=1) ## change

null.dat$UD50byarea <- mftable.new$UD50byarea

##

store[i,] <-summary(lmer(avg.deg~UD50byarea+(1|nest.x),data=null.dat))$coefficients[1:2]

print(i)

}

##

coef.mod <- summary(mod)$coefficients[1:2]

coef.mod

## p values

sapply(1:ncol(store),

function(x) length(store[abs(store[,x]) >= abs(coef.mod[x]),x]) / iter)

##############################################

#Bootstrap

iter <- 1000

store <- data.frame(intercept=numeric(iter), UD50byarea=numeric(iter))

for (i in 1:iter){ #> df[sample(nrow(df), 3), ]

pick.groups <- data.frame(nest.x = with(dat, sample(unique(nest.x), replace=TRUE)),

newcat = factor(1:length(unique(dat$nest.x)))) #Makes a new column so that number of groups is maintained if a group is dropped while resampling

boot.d <- dplyr::left\_join(pick.groups, dat) #Joins with whole dataframe

family="poisson",data=boot.d))$coefficients[1:4]

print(i)

store[i,] <-summary(lmer(avg.deg~UD50byarea+(1|newcat),data=boot.d))$coefficients[1:2]

}

store[1:2,]

out <- data.frame(sapply(1:ncol(store), function(x) quantile(store[,x],c(0.025,0.975))))

names(out) <- names(store)

out

mod<-summary(lmer(avg.deg~UD50byarea+(1|nest.x), data=dat))$coefficients[1:2]

cbind(estimate=mod, t(out))

mod<-lmer(avg.deg~UD50byarea+(1|nest.x), data=dat)

summary(mod)