Forage site distributions quantify collective foraging in honey bee colonies: Methods

Joseph Palmer, Ash Samuelson, Elli Leadbeater and Vincent Jansen

# Acknowledgments

# Conflict of interest

The authors declare no conflict of interest.

# Code and data availability

All code for analysis is available on GitHub at (insert link to finalised public github repo or archive). All data is available via the Dryad Digital Repository <https://doi.org/10.5061/dryad.c2fqz618f> (Samuelson, Schürch, and Leadbeater 2021).

# Data collection

Methods for how the data used for this study was collected can be found in full in Materials and Methods sections 2.1, 2.2 and 2.3 of (Samuelson, Schürch, and Leadbeater 2021).

## Land-use preference analysis

Methods for how land-use types were classified can be found in Materials and Methods sections 2.6 of (Samuelson, Schürch, and Leadbeater 2021).

## Waggle dance decoding

Methods for waggle dance decoding are fully described in Materials and Methods section 2.4 in (Samuelson, Schürch, and Leadbeater 2021).

# Simulation

A circular environment is first created with radius . The number of flowers in the environment is generated as a random Poisson variable with rate equal to 5000 multiplied by the area of the environment. These are placed on polar coordinates with a uniformly selected angle, , between 0 and 2 and a radial value, , between 0 and , determined from the square root of the uniform position values multiplied by . These polar coordinates are converted to Cartesian coordinates using the formulas and . Each location is then assigned to an instance of a flower object along with a random quality of between 0 and 10. This quality is combined with the distance of the flower to the centrally located hive to form a measure of how profitable the flower is (see model, equation?).

One-hundred honeybee objects are created, 20 of which 20 are scouts and the rest recruits. Scouts leave the environment following a random path through the environment generated by sampling a uniform random step length and angle. The number of paths the scout draws when searching is also determined as a uniform random number. Each straight line in the random path is converted to a rectangle with length equal to the path section length and a constant width of ~0.01 to represent an area the scout searches along that path. Of all the flowers contained in the boxes drawn from the scout’s path, the one closest to the colony is selected as the flower the scout will report and will communicate its location if the quality of the flower is 0. Flower communication is simulated by pooling together all the flower objects found. If no flowers are contained in the scout’s path, they will not add any flowers to the scout pool and draw a new path in the next foraging iteration.

Recruits represent honeybee objects which do not perform the searches the scouts do. Instead, they sample from the pool of flower objects reported by scouts. This sampling is done by selecting flowers with a probability which is skewed towards the profitability of the flower, meaning more profitable flowers have a greater chance of being selected by recruits. Recruits will then visit these flowers and in the next iteration will add their flower to the pool of scout dances. Consequently, the pool of dances represents flowers discovered by scouts and flowers exploited by recruits. When a flower is depleted,

the flower is removed from the environment and so any foragers that were foraging on it would select a different flower from the dance floor.

The simulation was run 100 times and every 5 time steps all distances reported by scouts and recruits were recorded and combined. This was done to reflect the way foraging data is collected in real honeybee studies. We fit an exponential and minimum of an exponential distribution to both the distribution of foraging distances reported by the scout and recruit objects. Fitting was done by deriving the maximum likelihood estimate for each model fit on each data source through their analytical solutions: , minimum of the exponential with a minimum foraging distance: . As the exponential assumes distributions start from 0 the data was transformed to start from 0 by subtracting the minimum foraging distance from all foraging distances () before fitting.

All simulation code was written in Python version 3.9 and uses the Pandas (McKinney 2011) and Scipy (Jones, Oliphant, and Peterson 2001) packages.

# Model

To describe the distribution of dance durations on the dance floor we formulated a generic model the duration of waggle dances. In the model resource patches are randomly placed in the environment. Foragers scout for these patches. The rationale of the model is illustrated in Fig. 3: upon visiting a resource patch, foragers translate the profitability of a resource into the number of repeats of the dance. The number of repeats of the dance is a function of quality an distance. Recruits sample random dances and report the location of successful visits to resource patches on the dance floor. Through the feedback and over-representation of profitable resources on the dance floor recruits will converge to visiting the most profitable resource in vicinity of the hive. The distribution of dance durations is the superposition of scouting and recruiting trips.

As the resources patches are randomly placed in the environment, the distance after which the first resource is discovered approximately follows an exponential distribution (given by ). Through the feedback mechanism that the dance floor provides, the colony can, collectively, locate the most profitable resource in its environment. For randomly placed resources in a two dimensional environment the distance to the nearest point is distributed according to a Rayleigh distribution (given by ) (Pyke 1978). Our simulation model shows that this describes the distances at which recruits visit resources well. Knowing the distance distributions of scout and recruit trips we then assume that the a proportion of all trips are scout trips. With this information we can specify the distributions of distances on the dance floor (see Supplementary Material for details).

We implemented this in a full model which describes the distance distribution of the environment has different resource types (See Supplementary Material). However, in the full model the number of parameters increases with . Even if the number of resources is low, it turned out to be cumbersome to estimate the parameters and the model tends to overfit. To facilitate estimation of the parameter we therefore used a simplified model to estimate the fraction of scout trips, where represents the lowest duration recorded, as there exists a minimum waggle run duration.

For the simplified the model we assumed that the the by assuming that the number of dances depends weakly on distance and there is a sizable quality differences between resources of a decent size and that there is a sizable intensity of the high quality resource. Foragers on scouting trips are more likely to report larger distances than foragers on recruiting trips. By linearising the function that translates the profitability into the number of waggle dance run for the largest distance and normalising, we arrive at simplified distribution for distances for scouting trips:

where we used the shorthand The parameter is the maximum reported distance by scouts, is the intensity of resources found by scouts and is factor that normalises the distribution.

Recruit trips will be predominantly to high quality resource. Only if the nearest high quality patch is very far away will there be a more profitable patch of lesser quality available, and this happens only rarely if the intensity of the best quality resource is sizable. After linearising the function that translates the profitability into the number of waggle dance run for short distances and normalising the distribution of distances reported from recruit trips in the simplified model is:

where

is the normalisation factor, the parameter is the rate with which dances repeats depends on distance for recruit trips and is the intensity of high quality resources reported by recruited foragers.

The simplified distribution function is

which we used for parameter estimation and model fittings.

# Statistical analysis

All analysis code is written in R (R Core Team 2020).

## Model fitting

All models are fit using Maximum likelihood estimation (Burnham and Anderson 2002) by summation of the log of the simplified distribution function outlined in the methods section: model. The numerical optimisation routine is written in c++ and uses the Nelder-Mead simplex algorithm (Nelder and Mead 1965) implemented in the ‘NLopt’ library (Johnson 2020) and interfaced to R (R Core Team 2020) using ‘Rcpp’ (Eddelbuettel and François 2011).

Model parsimony is assessed using Akaike information criterion (AIC) (Aho, Derryberry, and Peterson 2014) and Akaike weights. The model with the lowest AIC score is deemed to be the most parsimonious.

Goodness of fit is assessed using the Kolmorgorov-Smirnov (KS) test (Goldstein, Morris, and Yen 2004) and implemented in R using the ks.boot function of the package ‘Matching’ in R (Sekhon 2011).

## Partial Least Squares analysis

As our estimated proportion of scouts is continuos on the interval we performed a beta regression using the R package plsRbeta (Bertrand et al. 2013) to conduct the partial least squares analysis and performed a beta regression on the results using the R package betareg (Cribari-Neto and Zeileis 2010). As the betareg package only works on the open interval the data, , was transformed using the following equation: $, as outlined in the betareg package documentation. After analysis the data was back transformed to the original values for the plots in Fig 5.

For the jackknifed resampling we iterated through the each site and removed it from the pool of data and then ran the PLS as described above, recoding the loadings for each iteration (see Supplementary Material for loadings with each site removed). The PLS loadings for each land-use type are plotted as a box plot in Fig 5. to show the spread of these variable types.

# References

Aho, Ken, DeWayne Derryberry, and Teri Peterson. 2014. “Model selection for ecologists: the worldviews of AIC and BIC.” *Ecology* 95 (3): 631–36. <https://doi.org/10.1890/13-1452.1>.

Bertrand, Frédéric, Nicolas Meyer, Michèle Beau-Faller, Karim El Bayed, Izzie-Jacques Namer, and Myriam Maumy-Bertrand. 2013. “Régression Bêta PLS.” *Journal de La Société Française de Statistique* 154 (3): 143–59. <http://journal-sfds.fr/article/view/215>.

Burnham, K.P., and D.R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach.* 2nd ed. New York: Springer New York. <https://doi.org/10.1007/b97636>.

Cribari-Neto, Francisco, and Achim Zeileis. 2010. “Beta Regression in R.” *Journal of Statistical Software* 34 (2): 1–24. <https://doi.org/10.18637/JSS.V034.I02>.

Eddelbuettel, Dirk, and Romain François. 2011. “Rcpp: Seamless R and C++ integration.” *Journal of Statistical Software* 40 (8): 1–18. <https://doi.org/10.18637/JSS.V040.I08>.

Goldstein, M. L., S. A. Morris, and G. G. Yen. 2004. “Problems with fitting to the power-law distribution.” *The European Physical Journal B - Condensed Matter and Complex Systems 2004 41:2* 41 (2): 255–58. <https://doi.org/10.1140/EPJB/E2004-00316-5>.

Johnson, Steven G. 2020. “The NLopt nonlinear-optimization package.” <http://github.com/stevengj/nlopt>.

Jones, Eric; Travis; Oliphant, and Pearu; Peterson. 2001. “SciPy: Open source scientific tools for Python.”

McKinney, Wes. 2011. “pandas: a Foundational Python Library for Data Analysis and Statistics.” *Proceedings of the 9th Python in Science Conference*.

Nelder, J. A., and R. Mead. 1965. “A Simplex Method for Function Minimization.” *The Computer Journal* 7 (4): 308–13. <https://doi.org/10.1093/COMJNL/7.4.308>.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.r-project.org/>.

Samuelson, Ash E., Roger Schürch, and Ellouise Leadbeater. 2021. “Dancing bees evaluate central urban forage resources as superior to agricultural land.” *Journal of Applied Ecology*. <https://doi.org/10.1111/1365-2664.14011>.

Sekhon, Jasjeet S. 2011. “Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching package for R.” *Journal of Statistical Software* 42 (7): 1–52. <https://doi.org/10.18637/JSS.V042.I07>.