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

We make the initial assumption that there are different resources distributed within an environment according to a spatial Poisson point process with rate for resource . As a scout travels a small distance it will cover an area of , with depending upon the width of the area scanned, the detection rate and degree to which a scout’s path covers new ground. The probability density of the distance travelled after which the first resource is discovered follows an exponential distribution with a rate derived from the intensity of the flowers within the environment: , where .

The number of repeats of a dance on the dance floor is dictated by profitability and is defined as the net energy efficiency per amount of energy spent: . The gain is determined by the sucrose concentration of nectar from a flower, , whilst the cost is determined by the distance, . The energy gained, , is therefore proportional to the quality , whilst the cost increases linearly with distance. We can therefore refactor energy efficiency as , where represents the energy spend per distance relative to a fixed cost of travel. As bees should not be expected to dance for a negative profitability, we can derive a function to translate quality and distance into a measure of the number of dances on a dance floor: , where the notation of . This function has the property that for a fixed there is a minimum quality below which bees don’t dance. The equivalent quality of two resources at distance and is . The critical distance .

For a resource of type located at a distance , the expected number of dances would be . It follows that the expected total number of scout waggle dances on the dance floor is given by: . The distribution of scout dances is: .

Using the same logic we derive what the distribution of recruits will look like. If two locations differ in resource quality and distance then a hive can prefer a further location if it is sufficiently better and a lesser quality resource if it is sufficiently nearer. For a site with quality and at distance from the hive, for a second location with quality this threshold distance satisfies . Consequently, the critical distance is an implicit function of distance and it follows from implicit differentiation that and . If then and it follows that if then and if then . As recruits get their information from scouts (as well as recruits) how well the scouts cover depends upon the number of scouts and how well their collective paths cover an area and the chance of discovery of each resource. To capture this we let the fraction of resources that can be discovered by a collection of scouts be and assume . The probability that there is a not a food source discovered worth reporting on within a radius of , where . This function also described the probability of having the nearest food source that the recruits will come out for discovered at least a distance of away.

# 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 model 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 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

A key consideration is that our response variable is a proportion and thus lies on the interval . This makes it unsuitable for a standard linear regression. One option would be to use a generalized linear regression with a logit link and quasibinomial family. This method is most suited for when the response variable is an integer proportion, where the response variable is a division of two integers, such as the proportion of males within a population. Our measure is in principle an integer proportion as we measure the proportion of scouts, however, this value is an estimate derived from our model and so is continuous on the interval . An alternative method would be to transform the proportions on to a continuos scale using the transformation. The most appropriate method would be to use a beta regression which is designed specifically for proportional response variables.

We used 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).

# 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>.