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

# Acknowledgments

This work was supported by the Biotechnology and Biological Sciences Research Council (BBSRC) through grant BB/M011178/1.

# 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>1.

# 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 of1.

## Land-use preference analysis

Methods for how land-use types were classified can be found in Materials and Methods sections 2.6 of1.

## Waggle dance decoding

Methods for waggle dance decoding are fully described in Materials and Methods section 2.4 in1.

# Simulation

All simulation code was written in Python version 3.9 and uses the Pandas2 and Scipy3 packages.

A circular environment is first created with radius . The number of resources 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. Each location is then assigned to an instance of a resource object along with a random quality of between 0 and 10. This quality is combined with the distance of the resource to the centrally located hive to form a measure of how profitable the resource is (see model, equation?).

One-hundred honeybee objects are created, 20 of which 20 are on scouting trips and the rest recruited to follow scout dances. 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 resources contained in the boxes drawn from the scout’s path, the one closest to the colony is selected as the resource patch that the scout will report and will communicate its location if the quality of the resource resource exceeds a minimum threshold. Communication is simulated by pooling together all the resource patches found. If no resources are contained in the scout’s path, they will not add any resources 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 resource objects reported by scouts. This sampling is done by selecting resources with a probability which is skewed towards the profitability of the resource, meaning more profitable resources have a greater chance of being selected by recruits. Recruits will then visit these resources and in the next iteration will add their resource to the pool of scout dances. Consequently, the pool of dances represents resources discovered by scouts and resources exploited by recruits. When a resource is depleted, it is removed from the environment and so any foragers that were foraging on it would select a different resource 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.

# 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 assumed to be 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 ))4. 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 considered, and here a minimum waggle run duration in the data set.

For the simplified the model we assumed that the number of dances depends weakly on distance and there is a sizable quality differences between resources of a non-negligible size and that there is a sizable intensity of the high quality resource (See Supplementary Material for detailed derivation). 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 dance duration and normalising, we arrive at simplified distribution for dance durations for scouting trips:

where we used the shorthand The parameter is the maximum dance duration 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 resources. 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 runs for short dance durations and normalising the distribution of dance durations 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 R5.

## Model fitting

All models are fit using Maximum likelihood estimation6 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 algorithm7 implemented in the ‘NLopt’ library8 and interfaced to R5 using ‘Rcpp’9.

The most parsimonious model is assessed using Akaike information criterion (AIC)6,10 and Akaike weights. The model with the lowest AIC score is deemed to be the most parsimonious.

Goodness of fit is assessed using the two-sample Kolmorgorov-Smirnov (KS) test11 and implemented in R using the ks.boot function of the package ‘Matching’ in R12.

## Partial Least Squares analysis

Prior to conducing the PLS we removed any sites in which the models fit was significantly different to the waggle dance durations for that site. This resulted in one agri-rural site (ROT) and no urban sites being removed from the analysis.

As our estimated proportion of scouts is continuos on the interval we used the R package plsRbeta13 to conduct the partial least squares analysis and performed a beta regression on the results using the R package betareg14. 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. A loading was determined to be significantly correlating with the first principal component if contributed more than its expected variance.

# References

1. Samuelson, A. E., Schürch, R. & Leadbeater, E. Dancing bees evaluate central urban forage resources as superior to agricultural land. *Journal of Applied Ecology* (2021) doi:[10.1111/1365-2664.14011](https://doi.org/10.1111/1365-2664.14011).

2. McKinney, W. pandas: a Foundational Python Library for Data Analysis and Statistics. *Proceedings of the 9th Python in Science Conference* (2011).

3. Jones, E., Oliphant, T. & Peterson, P. SciPy: Open source scientific tools for Python. (2001).

4. Pyke, G. H. Optimal foraging in bumblebees and coevolution with their plants. *Oecologia 1978 36:3* **36**, 281–293 (1978).

5. R Core Team. *R: A Language and Environment for Statistical Computing*. (R Foundation for Statistical Computing, 2020).

6. Burnham, K. & Anderson, D. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach.* (Springer New York, 2002). doi:[10.1007/b97636](https://doi.org/10.1007/b97636).

7. Nelder, J. A. & Mead, R. A Simplex Method for Function Minimization. *The Computer Journal* **7**, 308–313 (1965).

8. Johnson, S. G. The NLopt nonlinear-optimization package. (2020).

9. Eddelbuettel, D. & François, R. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software* **40**, 1–18 (2011).

10. Aho, K., Derryberry, D. & Peterson, T. Model selection for ecologists: the worldviews of AIC and BIC. *Ecology* **95**, 631–636 (2014).

11. Goldstein, M. L., Morris, S. A. & Yen, G. G. Problems with fitting to the power-law distribution. *The European Physical Journal B - Condensed Matter and Complex Systems 2004 41:2* **41**, 255–258 (2004).

12. Sekhon, J. S. Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching package for R. *Journal of Statistical Software* **42**, 1–52 (2011).

13. Bertrand, F. *et al.* Régression Bêta PLS. *Journal de la Société Française de Statistique* **154**, 143–159 (2013).

14. Cribari-Neto, F. & Zeileis, A. Beta Regression in R. *Journal of Statistical Software* **34**, 1–24 (2010).