Supporting Information

bmotif: a package for motif analyses of bipartite networks

Benno I. Simmons[1], Michelle J. M. Sweering[1,2], Maybritt Schillinger,[1,2] Lynn V. Dicks[1,3], William J. Sutherland[1], Riccardo Di Clemente[4,5]

[1] Conservation Science Group, Department of Zoology, University of Cambridge, The David Attenborough Building, Pembroke Street, Cambridge CB2 3QZ, UK

[2] Faculty of Mathematics, Wilberforce Road, Cambridge CB3 0WA, UK

[3] School of Biological Sciences, University of East Anglia, Norwich NR4 7TL, UK

[4] Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Massachusetts Avenue 77, MA 02139, Cambridge, USA

[5] Centre for Advanced Spatial Analysis (CASA), University College London, Gower Street, London, WC1E 6BT, UK

Corresponding authors:

Benno I. Simmons. Address: Conservation Science Group, Department of Zoology, University of Cambridge, The David Attenborough Building, Pembroke Street, Cambridge, CB2 3QZ, UK. Email: benno.simmons@gmail.com

Riccardo Di Clemente. Address: Centre for Advanced Spatial Analysis (CASA), University College London, Gower Street, London, WC1E 6BT, UK. Email: r.diclemente@ucl.ac.uk

Appendix S4: Computational performance

Empirical networks

To assess the speed of bmotif functions, we used *mcount* and *node_positions* to calculate the complete motif profiles of 175 empirical pollination and seed dispersal networks and the positions of all their constituent species. Networks were obtained from the Web of Life dataset (www.web-of-life.es). The networks varied in size from 6 to 797 species (mean: 77.1; standard deviation: 117.8). Analyses were carried out on a computer with a 4.0 GHz processor and 32 GB of memory. Functions were timed using the R package 'microbenchmark' (Mersmann, 2015). Results are shown in Fig. S2.

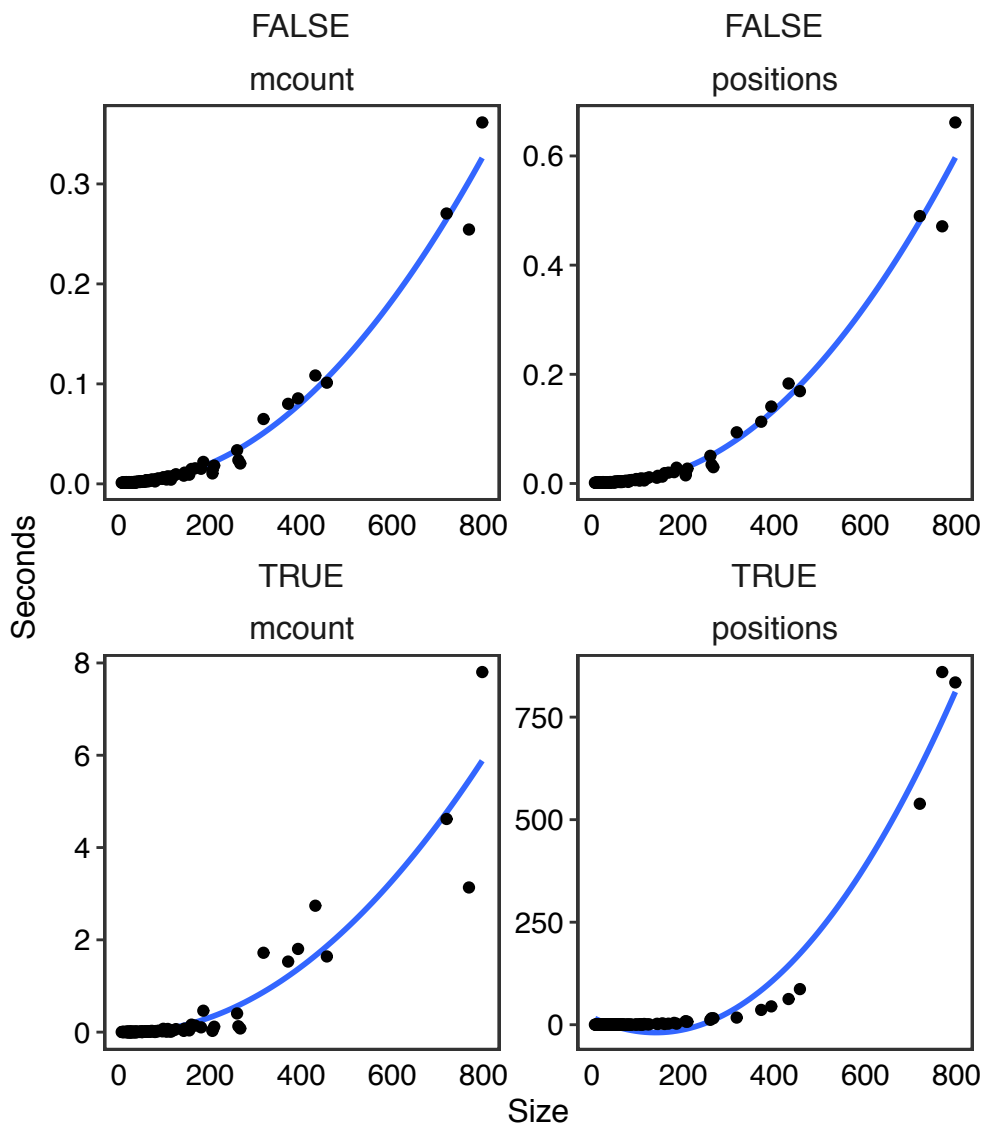

Figure S2: Relationship between network size and computational performance for *mcount* and *node_positions* for motifs containing up to five nodes (FALSE) and six nodes (TRUE). Functions were timed on 175 empirical networks. Lines are best fit polynomial curves of degree 2.

As expected, the time taken for a function to run increases monotonically with the size of the network (number of species). When six-node motifs were excluded, *mcount* and *node_positions* took 0.36 and 0.66 seconds, respectively, to complete for the largest network

in our dataset (797 species). For smaller networks which are more typical of the communities analysed by ecologists, both functions completed in substantially less than one second. This speed is possible as all formulae involved in calculations of motifs up to five-nodes use relatively simple operations, such as matrix multiplication or the binomial coefficient. When six-node motifs were included, for a network with 78 species (close to the mean network size of 77.1 species), *mcount* completed in 0.01 seconds, while *node_positions* completed in 0.32 seconds. For the largest network, *mcount* completed in 7.8 seconds, while *node_positions* took 13.9 minutes. Six-node motifs slow down calculations as, unlike five-node motifs, their algorithms require the use of the tensor product. Overall, the speed of bmotif makes motif analyses compatible with the permutational approaches frequently used in network ecology, particularly for analyses with motifs up to five-nodes and for six-node analyses of all but the largest networks. For example, using bmotif it would be feasible to calculate motif frequency distributions across thousands of null networks, which is a widely-used approach to disentangle the mechanisms responsible for network structure (Bascompte, Jordano, Melián, & Olesen, 2003; Dormann, Frund, Bluthgen, & Gruber, 2009).

Random networks

We carried out two analyses using randomly-generated networks to examine the effects of network size (number of species) and connectance on the computational performance of individual motif and motif position calculations. For the first analysis, we generated random networks with a fixed size, varying the connectance between 0.2 and 1. We generated 1000 networks for each value of connectance. For each of these sets of 1000 networks, we recorded the mean time for our code to calculate the frequency of five motifs (motifs 1, 2, 5, 10 and 28; one from each of the five motif size classes) and the number of times each species occurred in five motif positions (positions 1, 3, 9, 23 and 85; one from each motif size class). The dimensions of the generated networks were set as the median number of rows and columns of 230 empirical ecological bipartite networks (22 rows, 13 columns) obtained from the Web of Life repository (www.web-of-life.es). For the second analysis, we generated random networks of a fixed connectance, varying the size between 10 and 200 species. We generated 1000 networks for each value of size and recorded the mean time for our code to calculate the frequency of the same five motifs and positions. The connectance of the generated networks was the median connectance of the empirical network dataset (0.243) and the row:column ratio (ratio of number of species in one level, such as hosts, to the number of species in the other level, such as parasitoids) was also set as the empirical median (2). Functions were timed using the R package 'microbenchmark' (Mersmann, 2015).

We found that connectance had little effect on the performance of individual motif and position calculations (Fig. S3), while a polynomial of degree two explained the increase in time with network size ($R^2 > 0.99$ for all motifs and positions) (Fig. S4).

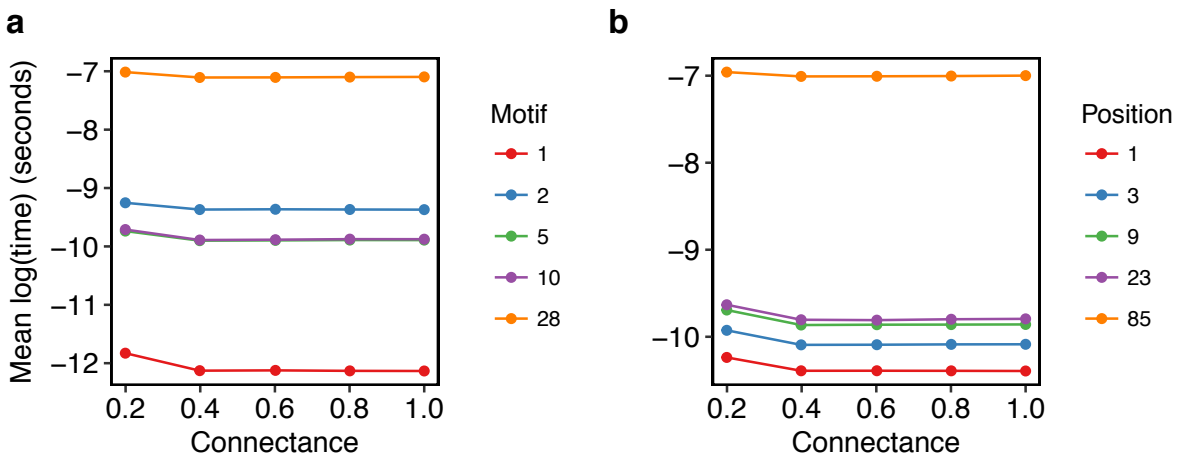

Figure S3: Relationship between connectance and computational time taken to calculate the frequency of (a) five motifs, one from each motif size class, and (b) five motif positions, one from each motif size class. Functions were run on randomly-generated networks of a given connectance. For each level of connectance, we generated 1000 random networks and record the mean time for the functions to complete. Lines connecting each point are shown for visualisation.

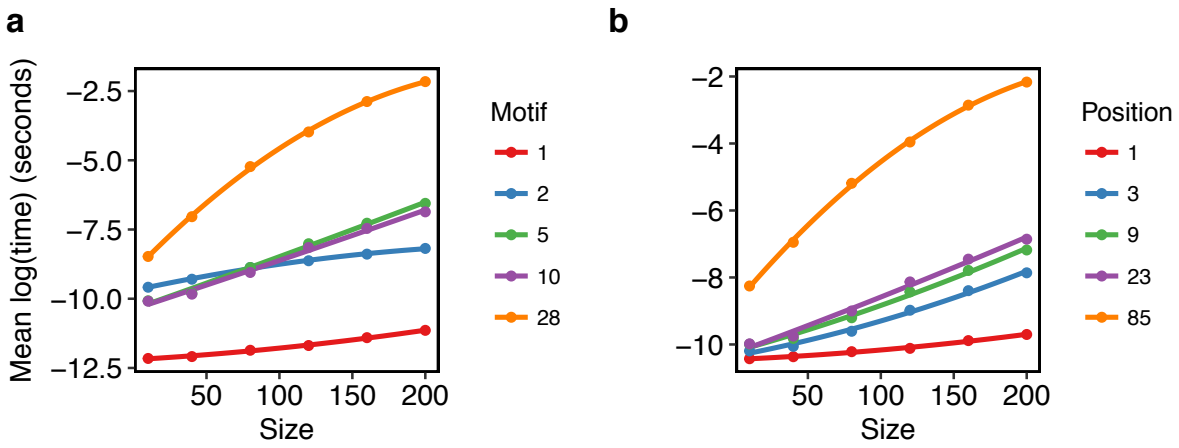

Figure S4: Relationship between size and computational time taken to calculate the frequency of (a) five motifs, one from each motif size class, and (b) five motif positions, one from each motif size class. Functions were run on randomly-generated networks of a given size. For each level of size, we generated 1000 random networks and record the mean time for the functions to complete. Lines are best fit polynomials of degree two.

Appendix S2: Matrix representation of motifs

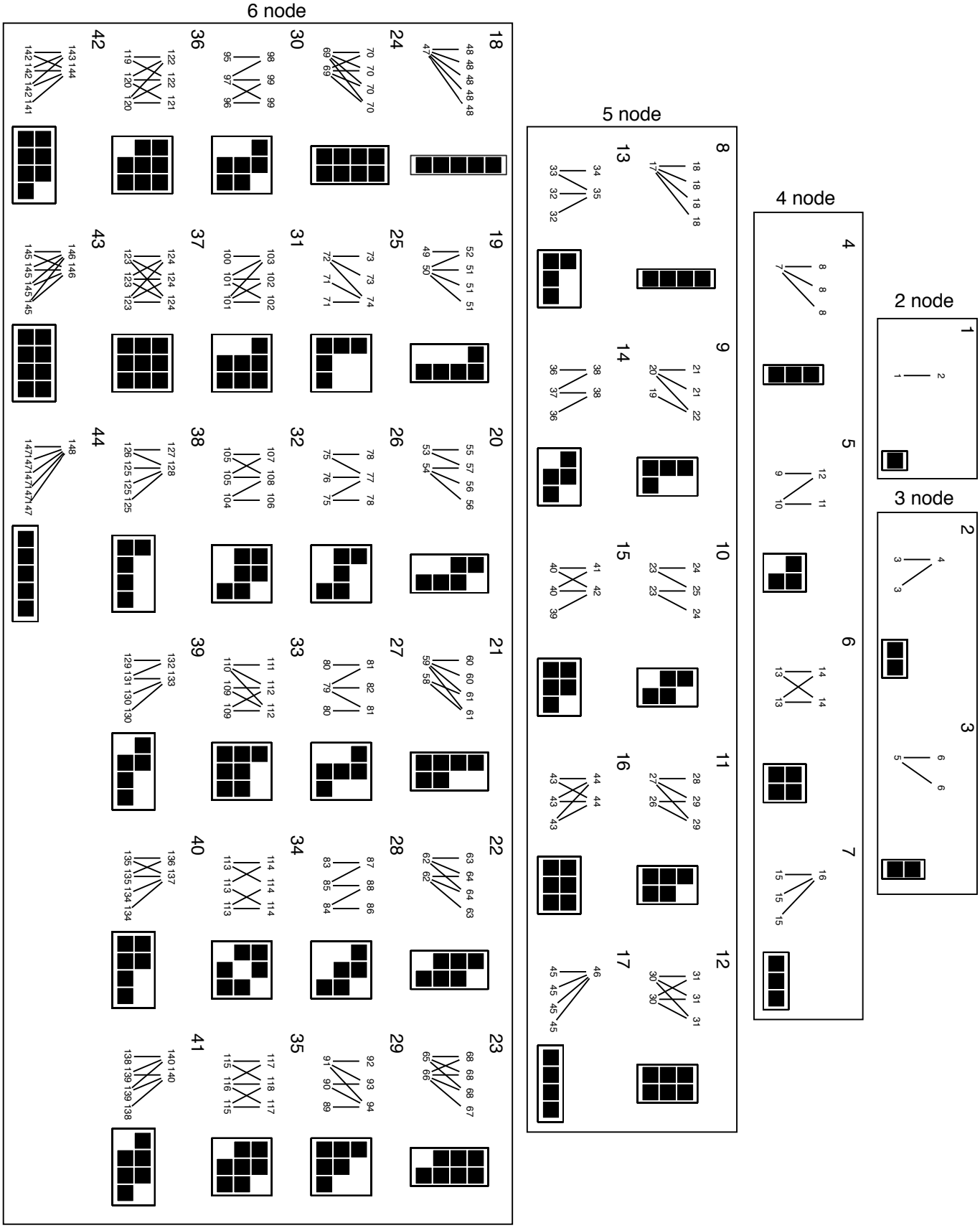

Figure S1: All bipartite motifs containing up to 6 nodes (species) and their corresponding representation as biadjacency matrices. Large numbers identify each motif. Small numbers represent the unique positions species can occupy within motifs, following Baker et al. (2015) Appendix 1. Lines between small numbers indicate undirected species interactions. To the right of each motif is its corresponding biadjacency matrix, **M**: black squares indicate a 1 in the matrix (the presence of an interaction), white squares indicate 0 (the absence of an interaction). There are 44 motifs containing 148 unique positions..

Appendix S5: Description of *mcount* and *node_positions* outputs

mcount takes a network as input and returns a data frame with one row for each motif (17 or 44 rows depending on whether motifs up to five or six nodes are requested, respectively) and three columns. The first column is the motif identity as in Fig. 1; the second column is the motif size class (number of nodes each motif contains); and the third column is the frequency with which each motif occurs in the network (a network's motif profile). For comparing multiple networks it is important to normalise motif frequencies. Therefore, if the 'normalisation' argument is TRUE, three columns are added to the data frame, each corresponding to a different method for normalising motif frequencies. The first column ('normalise_sum') expresses the frequency of each motif as a proportion of the total number of motifs in the network. The second column ('normalise_sizeclass') expresses the frequency of each motif as a proportion of the total number of motifs within its size class. The final column ('normalise_nodesets') expresses the frequency of each motif as the number of species combinations that occur in a motif as a proportion of the number of species combinations that could occur in that motif. For example, in motifs 9, 10, 11 and 12, there are three species in the top set (A) and two species in the lower set (B) (Fig. 1). Therefore, the maximum number of species combinations that could occur in these motifs is given by the product of binomial coefficients, choosing three species from A and two from P: $\binom{A}{3}\binom{B}{2}$ (Poisot & Stouffer, 2016). The most appropriate normalisation depends on the question being asked. For example, 'normalise_sum' allows for consideration of whether species are more involved in smaller or larger motifs. Conversely, 'normalise_sizeclass' focuses the analysis on how species form their interactions among different arrangements of n nodes.

node_positions takes a network as input and returns a data frame, **W**, with one row for each species and one column for each node position (46 or 148 columns, depending on whether motifs up to five or six nodes are requested, respectively; Fig. 1). w_{rc} gives the number of times species r occurs in position c. Each row thus represents the structural role or 'interaction niche' of a species. The 'level' argument allows positions to be requested for all species, species in set A only or species in set B only, returning a data frame with $A + B$ rows, A rows or B rows, respectively. Two types of normalisation are provided: 'sum' normalisation expresses a species' position frequencies as a proportion of the total number of times that species appears in any position; 'size class' normalisation uses the same approach, but normalises frequencies within each motif size class. Again, the most appropriate normalisation depends on the question being asked: if movements between motif size classes are of interest, 'sum' normalisation is most appropriate; if the focus is on how species form interactions among a given number of nodes, then 'size class' normalisation should be chosen.

References

Bascompte, J., Jordano, P., Melián, C. J., & Olesen, J. M. (2003). The nested assembly of plant-animal mutualistic networks. *Proceedings of the National Academy of Sciences*, *100*(16), 9383–9387.

Dormann, C. F., Frund, J., Bluthgen, N., & Gruber, B. (2009). Indices, Graphs and Null Models: Analyzing Bipartite Ecological Networks. *The Open Ecology Journal*, *2*(1), 7–24. doi:10.2174/1874213000902010007

Mersmann, O. (2015). microbenchmark: Accurate Timing Functions. R package version 1.4-

2.1.
Poisot, T., & Stouffer, D. (2016). How ecological networks evolve. *BioRxiv*. Retrieved from http://biorxiv.org/content/early/2016/08/29/071993.abstract