

okaapi: an R package for generating social networks based on trait preferences

Josefine Bohr Brask^{1,2,3}, Lauren JN Brent⁴, Delphine De Moor^{4,5}

1. Copenhagen Center for Social Data Science (SODAS), University of Copenhagen, Copenhagen, Denmark
2. Section for Ecology and Evolution, Department of Biology, University of Copenhagen, Copenhagen, Denmark
3. DTU Compute - Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Kongens Lyngby, Denmark
4. Centre for Research in Animal Behaviour, University of Exeter, Exeter, UK
5. Department of Primate Behavior and Evolution, Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany

* Corresponding author. Email: bohrbrask@gmail.com

Abstract

The study of simulated networks plays a central role for our understanding of the emergence and consequences of network structures. Accessible software to create simulated networks based on relevant processes can facilitate the use of this important approach in behavioural ecology, and drive forward our understanding of animal social structures.

Here we present the R package ‘okaapi’. This package can create simulated networks based on a key driver of real social networks, namely trait preferences:behaviour where individuals differentially socialize with others that have certain trait values, such as old age or the same sex as oneself.

The okaapi package provides tools for generating, visualising and quantifying trait preference networks. It uses a flexible modelling approach, where many different trait preferences can be modelled, and multiple trait preferences can affect the network simultaneously (as in real networks). It can both generate purely simulated networks, and networks based on trait data from real populations.

The package can also be used for modelling networks with other node effects than trait preferences, and may be useful not only for research in animal social networks, but also human social networks and non-social networks.

This paper provides an introduction to the okaapi package, including information on the kinds of studies where it may be useful, its framework and functionality, its content, and examples of its use (with accompanying R code).

We hope that the okaapi package will be useful in the field of behavioural ecology and other research areas, and that it will help facilitate the use of generative network modelling in the study of animal social systems and further the integration of this approach with empirical approaches.

Keywords

Generative network models, network modelling, network simulation, node attributes, R package, social networks, social preferences, trait preferences

1. Introduction

Social network structures are fundamental components of social systems, both in humans and non-human animals. The study of social networks (patterns of social relationships) in different species is important for understanding the interconnection between ecological and social environments and the evolution of social behaviour.

Generative network models are a key tool for understanding social networks. These models (in the sense we consider them here) are essentially algorithms that build simulated networks, based on rules for how network nodes (for example individuals) link to each other (Newman et al. 2018, Brask et al. 2025).

Generative network models have been used widely in network science and play a central role for our understanding of the drivers and consequences of network structures (for an overview see Brask et al. 2025). In the field of behavioural ecology, generative network models have been used for a range of purposes in relation to animal social networks (reviewed in Brask et al. 2025), including investigating how underlying generative processes affect social network properties (e.g. Ilany & Akcay 2016, Cantor & Farine 2018, Brask et al. 2026), how structural features of social networks affect network transmission and robustness (e.g. Sah et al. 2017, Romano et al. 2018, Evans et al. 2020, Cantor et al. 2021, Brask & Brask 2024), and how sampling biases affect inferred social network structures (e.g. Franks et al. 2010, Silk et al. 2015, Farine & Strandburg-Peskin 2016; Weiss et al. 2021).

Despite the usefulness of generative network models for animal social networks research, accessible tools for network generation based on relevant generative processes have been very scarce (exceptions include Silk & Gimenez 2023, Ross et al. 2024). Accessible tools that can generate networks based on processes that drive the real networks are essential for a more widespread use of generative network modelling in the field of behavioural ecology.

Here, we present the R package ‘okaapi’, which contains easy-to-use tools for generating, visualising and quantifying networks that are based on a key generative process of real social networks: *trait preferences*.

Trait preferences are behaviours where individuals differentially socialize with others that have certain trait values, such as their age, sex, body size, etc. Empirical research has shown that trait preferences play a central role in the structuring of social networks, across species from insects to humans, birds, fish, etc. (reviewed in Brask et al. 2026), and research questions about social networks often concern these preferences (De Moor et al. 2025).

The okaapi package provides a general and flexible method for generating networks based on this key real-world generative process, where multiple traits can simultaneously affect the network via different preferences, as is the case in the real world (Brask et al. 2026). The package complements existing tools for network generation based on other generative processes, developed for behavioural ecology (Silk & Gimenez 2023, Ross et al. 2024) or network science more broadly (e.g. igraph, Csárdi & Nepusz 2006). The package can be used for studies of animal social networks in any species, and may also be useful for research in human social networks, non-social networks, and theoretical networks that do not represent a specific system.

We hope that the okaapi package can help facilitating the use of generative network modelling in behavioural ecology and beyond, and lead to new insights in social and other networked systems.

2. Studies where the okaapi package may be useful

The most obvious use of the okaapi package may be for studies that concern trait preferences; for example studies investigating how different trait preferences affect network structure and function (as in Brask et al. 2026). Given the widespread importance of trait preferences (Brask et al. 2026), such studies could be relevant for many different species (including humans).

The package may also be used in studies where the network structures that okaapi can create are in themselves useful regardless of the generative processes behind them; for example simulation studies that investigate the transmission of disease and information in networks with different structural properties.

An additional use of the package is in studies where nodes affect the networks in other ways than via trait preferences. This is possible because the trait preferences modelled by okaapi can be interpreted as other node attribute effects that are technically equivalent to trait preferences. For example, if the size and geographical position of airports affect their chance of them being linked by a flight route, then the airport network could be modelled with okaapi. Given that okaapi can model many different trait preferences, it can also be used to model many different node attribute effects.

3. Introduction to the okaapi package

The purpose of the okaapi package is to provide accessible tools for the generation, visualisation and investigation of networks that are based on trait preferences (or node attribute effects more generally, Section 2). The name ‘okaapi’ stands for ‘generation of networks based on social preferences for traits’, and is inspired by the okapi (a rare and elusive mammal also known as the zebra giraffe; Mallon et al. 2015). The package uses the trait preference model to generate simulated networks (see Brask et al. 2026 for general information about the model and mathematical description of it). The okaapi package is currently available for R.

In addition to the package, several accompanying resources are also available (Table 1). These include code examples, relevant papers, and a user guide. The guide contains important information about how to use the package in practice (including for node effects other than trait preferences), and will be continuously updated in case of new versions of the package. We therefore highly recommend reading the guide before using the package.

Table 1. An overview of current resources related to the okaapi R package

Resource	Description	Location
okaapi R package	A set of tools for network generation based on trait preferences, as well as network visualisation and network quantification.	See Code availability statement
okaapi function R help pages, incl. brief code examples	R help pages with detailed information about each function in the okaapi package, including function arguments, function output, and brief code examples showing how to use the function.	Within the okaapi package (accessed the same way as any R help page)
okaapi demonstration R code	A code providing a brief demonstration of how to use the main functions of the okaapi package.	See Code availability statement
okaapi examples R code	A code providing examples of using the okaapi package for research. Accompanies Section 5 of this paper.	See Code availability statement
Example R code from generative network models introduction paper	Code examples accompanying a paper with case studies that use okaapi. Note: the original purpose of these examples was to demonstrate generative network modelling rather than okaapi in particular.	See Brask et al. 2025
okaapi user guide	A guide providing information on how to use the okaapi package in practice.	See Supplementary material statement
okaapi introduction paper	A paper providing a general introduction to the okaapi package.	In front of you
Trait preference model paper	A paper introducing the generative network model that the okaapi package uses for network generation (the trait preference model) and uses it to investigate the effect of trait preferences on network structure and function.	Brask et al. 2026

3.1. The trait preference framework used in okaapi

To understand how the okaapi package works, it is helpful to be familiar with the terminology and definitions for trait preferences used in the package. They follow that of the underlying generative network model (the trait preference model presented in Brask et al. 2026; details and mathematical description of trait preferences can be found there).

3.1.1. Trait preferences

In okaapi, the term *trait preference* refers to behaviour where individuals differentially socialize with others that have certain trait values (such as their sex, age, body size, etc.). The term covers both active preferences, where individuals actively use the traits of others in their decisions of who to socialize with, and passive preferences, where the trait-based socializing is due to other factors than active choice (such as trait-dependent habitat preferences; Brask et al. 2026). This broad definition is used because active and passive preferences can be modelled equivalently in okaapi. A trait preference consists of a *trait type* combined with a *preference type*.

3.1.2. Trait types

In okaapi, a *trait type* refers to a specific distribution of trait values, such as categorical or normally distributed. A given trait type may be used to model any real-world trait that fits with that distribution. A categorical trait may for example be used to model sexes or colour morphs, a normally distributed trait may for example be used to model body size, and equally distributed integers may for example be used to model social ranks.

3.1.3. Preference types

In okaapi, a *preference type* is a specification of which trait values are preferred over others (irrespective of the specific trait). Preference types belong to two general types: *Similarity preferences* are preferences where individuals' preference for trait values depends on how similar they are to their own value of the trait. This includes preference for others that are similar to oneself (such as similar age), as well as preference for others that are dissimilar, moderately similar, etc. *Popularity preferences* are preferences where individuals prefer certain trait values regardless of their similarity to their own value. This includes preference for others that have high trait values (such as old age), as well as preference for others that have low trait values, average trait values, etc.

3.1.4. Combining trait types and preference types to get trait preferences

To model a trait preference in okaapi, the user combines a trait type with a preference type. For example, preference for socializing with one's own sex may be modelled by combining a categorical trait with a similarity preference.

3.2. The functionality of okaapi in a nutshell

The okaapi package can both create fully simulated networks (i.e. without using any empirical data), and simulated networks based on empirically measured trait values (such as sexes, ages, etc. from real populations).

Network generation in okaapi consists of three steps: first it creates trait values for all the individuals based on trait types set by the user (unless the trait values are input by the user); then it calculates a *social attraction value* for each pair of individuals, based on their trait values and preference types set by the user; then it determines the positions and weights of network links based on the social attraction values, such that pairs with higher social attraction are more likely to get a link and the link is likely to be stronger (see Brask et al. 2026 for details of the network generation mechanisms).

Key features of okaapi functionality include the following: Firstly, it can generate networks based on a range of different trait preferences: it includes both similarity and popularity preference types, and each of these can be combined with any trait type. Secondly, it can generate networks based on no trait preference (random social linking), a single trait preference, or multiple simultaneously acting trait preferences, reflecting that several traits are often simultaneously of importance in real social networks (Brask et al. 2026); for example, individuals may both prefer to socialise with individuals of their own sex, and with older individuals, with their social connections being based on a combination of these preferences. Thirdly, the trait preferences may be of different importance, both in the sense that trait preferences acting simultaneously do not need to have the same importance (reflecting real-world networks where one trait may be more important than another for social connection), and in the sense that the trait preferences may overall have more or less importance compared to random linking.

The okaapi package thus provides a flexible method for modelling trait preference networks, where the networks may be based on different trait preferences, which may act alone or together, and may have different importance.

3.3. Network structures generated by okaapi

Different trait preferences (and their combinations) give rise to different structures, and the okaapi package can therefore create a wide range of network structures (examples in Fig. 1). This includes networks with different structural characteristics such as modularity and centralization; with trait-based patterns such as trait assortment and correlations between trait values and individual social centrality; and with increases or decreases in network metrics such as clustering, path length, and degree variation (see Brask et al. 2026 for how specific trait preferences affect some of these structural properties). It can also create networks across the range from random to highly structured, and across the range from being dominated by one structural property to another (Fig. 1).

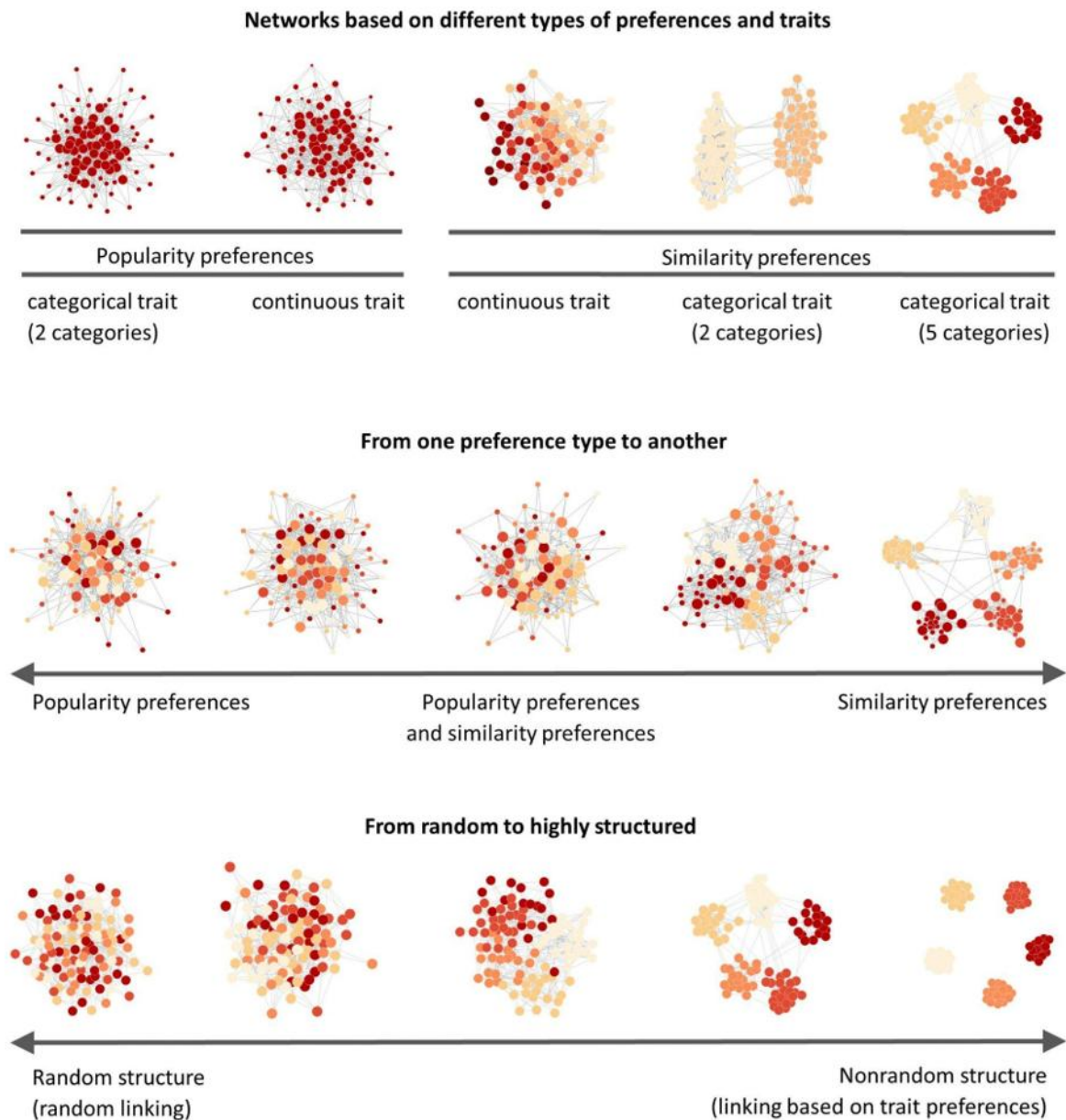


Figure 1. Examples of networks generated with the okaapi package. The package can create networks based on different preference types, which can each be combined with different trait types (upper row); it can create networks based on a single or multiple trait preferences, which can be of different relative importance (middle row); and it can create networks from trait preferences that are overall of more or less importance compared to randomness (bottom row). It can thus create a wide range of network structures, and it is only possible to show a limited set of examples here. Node sizes correspond to trait values from traits used with popularity preferences, and node colours correspond to trait values from traits used with similarity preferences.

4. Contents of the okaapi package

The okaapi package currently contains eight functions. An overview of the functions is given in Table 2, and the package structure is shown in Figure 2. Below we provide general descriptions of the functions. Details about function arguments and output is given in their help pages (see Table 1). We encourage researchers wanting to use the functions to first consult the okaapi user guide (Table 1). Any functions added to okaapi in the future will also be described in the guide.

Table 2. An overview of functions available in the okaapi package (at the time of writing). The two main functions provide tools for generating, visualising and quantifying trait preference networks. The helper functions used by the main functions. They may occasionally be relevant to use as stand-alone functions and are therefore available to users, but note that their use requires special attention.

	Function name	Function purpose
Main functions	<i>traitnet</i>	Generate and plot a trait preference network
	<i>traitnetsmetrics</i>	Generate an ensemble of trait preference networks and measure network metrics on them
Helper functions	<i>traitvalues</i>	Generate trait values
	<i>traitnetsociat</i>	Generate social attraction values based on trait preferences
	<i>traitnetbuild</i>	Generate a network matrix based on social attraction values
	<i>contincols</i>	Generate node colours based on values from a continuous scale
	<i>traitnetvisual</i>	Plot a trait preference network
	<i>netmetrics</i>	Measure network metrics on a network

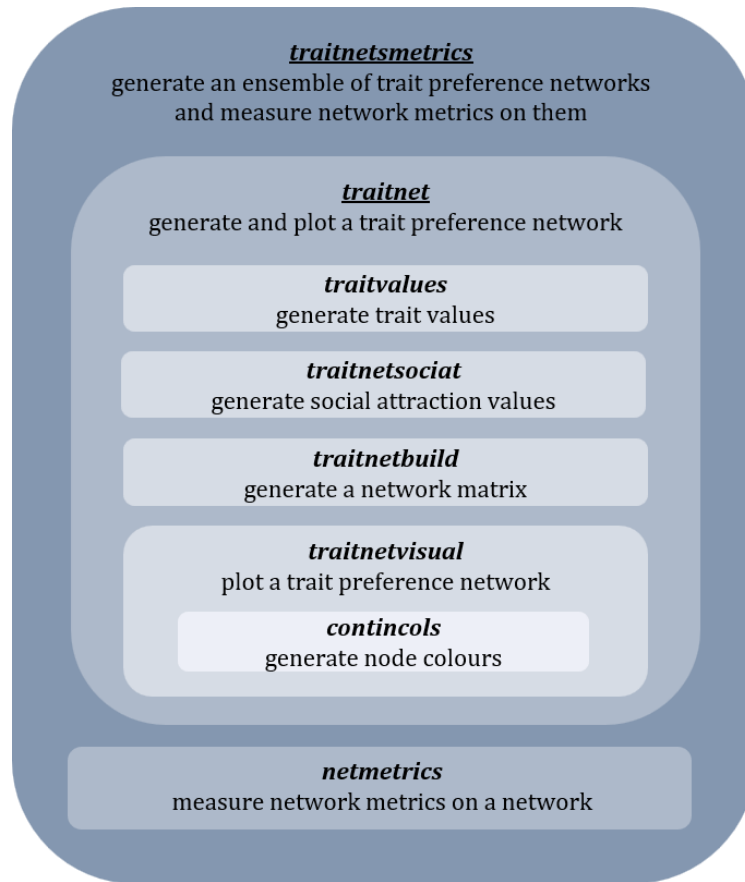


Figure 2. Structure of the okaapi package. The figure shows which okaapi functions use each other. Functions that are used by another function are nested within that function in the figure. For example, the *traitnetsmetrics* function uses the *traitnet* function and the *netmetrics* function.

4.1. The *traitnet* function

The first main function, *traitnet*, creates a trait preference network (i.e. an adjacency matrix containing the link weight for each pair of individuals) and - if desired - a plot of the network. It can also generate networks with an entirely random structure.

The *traitnet* function contains a number of parameters the user can set (function arguments), concerning the network and the trait preferences the network should be based on. The network parameters include the network size (number of nodes), average degree (average number of links per node), link type (various unweighted and weighted options), whether the network should have a single component (i.e. all nodes are at least indirectly connected), and whether the network should be plotted. The trait preference parameters include the trait type(s), number of categories (for categorical traits), preference type(s), importance(s), and (when relevant) values and classes of user-provided traits.

As output, the function gives the network adjacency matrix and (if desired) the network plot, together with some additional information (including the trait values of all individuals and the social attraction values (Section 3.2) of all pairs of individuals).

4.2. The *traitnetsmetrics* function

The second main function, called *traitnetsmetrics*, quantifies network metrics (structural measures) on an ensemble of trait preference networks (multiple networks generated from the same set of parameters). This reflects a standard approach in studies that involve simulated networks, where researchers work with a large set of networks of a given type, which may be considered as corresponding to replicates in an experimental treatment group (Brask et al. 2025).

The *traitnetsmetrics* function uses the *traitnet* function to generate each network. The *traitnetsmetrics* function therefore includes all the parameters (function arguments) of the *traitnet* function (Section 4.1), except for whether the network should be visualised (this is inconvenient for the purpose of *traitnetsmetrics* and therefore suppressed). It furthermore includes three additional parameters: The number of replicates (i.e. the number of networks in the ensemble), the metrics that should be measured, and whether the current replicate number should be written to the console (this is to enable the user to follow the progression of the task, as this may take more than an instant, depending on e.g. the network size and number of replicates).

The function output is the metric measurements for all the network replicates (for the sake of efficiency, the network matrices are not retained).

4.3. The helper functions

The helper functions perform different parts of the network generation and metric quantification, and are used by the main functions. They can also be used separately, but they should in this case be used with caution due to requirements of their arguments, which must fit correctly together (for details see the help pages of the helper functions in R).

Of the helper functions, the *traitvalues*, *netmetrics*, and *contincols* functions may be particularly useful as stand-alone functions. The *traitvalues* function can be used to create sets of trait values (or more generally, node attributes or values drawn from different distributions). The *netmetrics* function can measure network metrics on an input network that does not have to be generated using the *okaapi* package. The *contincols* function can create colours from a continuous colour scale, which can be used to colour a network, and it is particularly useful for colouring networks visualised with the *igraph* R package (Csárdi & Nepusz 2006).

5. Examples of using the okaapi package

Here we provide examples of using the okaapi package for scientific investigations, with accompanying R code (see Code availability statement). A code with a simpler demonstration of how to use the functions is also available (Table 1). For researchers wanting to use okaapi, we suggest also looking at the user guide and the function help pages (these contain important information about how to use the functions), the demonstration code, and potentially other okaapi resources (Table 1).

Below we show two examples: using okaapi 1) to explore trait preference networks visually, and 2) to measure structural properties (network metrics) of trait preference networks. The accompanying code includes code for these examples, as well as two additional examples: measuring aspects of trait preference networks with measures that are not currently implemented in okaapi, and generating & measuring networks using user-provided trait values.

For the examples, we focus on two trait preferences: a preference for socializing with individuals of one's own sex, modeled as a similarity preference combined with a categorical trait with two categories, and a preference for socializing with individuals that have a large body size, modeled as a popularity preference combined with a continuous, normally distributed trait. In the following, we refer to these two trait preferences as the *sex similarity preference* and the *size popularity preference*, respectively. Our aim is to gain insights into how these two trait preferences affect the structure of social networks.

5.1. Example: Exploring trait preference networks visually

As a first step to understand the effect of the two preferences on network structure, we want to explore what networks based on the preferences look like. While not all structural features are easily discernible by the human eye, this can give us a general impression of the structures that arise from the preferences.

To do this, we use the *traitnet* function to create and visualise trait preference networks. Our approach is to set the function arguments according to the two trait preferences, generate networks with varying importance of each preference, and look at the network plots (see the accompanying code for details).

Fig. 3 shows some of the generated networks. The exploration suggests that: 1) the sex similarity preference leads to a modular structure with a module (network community) for each sex (Fig. 3a), 2) the size popularity preference leads to a centralized structure with individuals of large body size being central (Fig. 3e), and 3) both preferences acting together leads to networks with a mix of these two structural features (Fig. 3b-d).

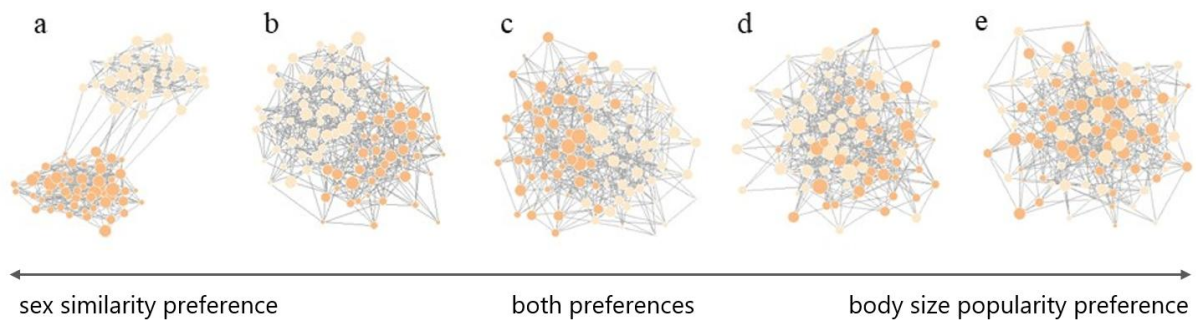


Figure 3. Visual exploration of networks based on sex similarity and body size popularity trait preferences. The networks were generated and plotted with the *traitnet* function. The colours correspond to sexes, and the node sizes correspond to body size. The networks are based on either preference alone (a, e), or both together with varying relative importance (b-d). The importance values were as follows: a: sim 0.95, pop 0.00. b: sim 0.60, pop 0.30. c: sim 0.50, pop 0.50. d: sim 0.30, pop 0.60. e: sim 0.00, pop 0.95.

5.2. Example: Measuring structural aspects of trait preference networks

Next, we want to measure structural aspects of networks that are based on the two trait preferences, to quantify the effect of the preferences on network structure. (Such an investigation could also be useful for investigating whether networks generated with specific settings have desirable structural properties, e.g. for using them to study how network structure affects spreading processes).

We use the *traitnetsmetrics* function to generate network ensembles and calculate metrics on them. We are interested in the effect of the preferences alone and together, and we therefore choose to make four network ensembles: One for just the sex similarity preference, one for just the size popularity preference, one for both preferences together, and one for no preferences (i.e. random networks, to have a baseline for comparison).

We set the function arguments according to the two preferences (as above), and we also set arguments for the number of networks we want in each ensemble and the metrics we want to calculate (see code for details). To make the four ensembles, our approach is to write the function call one time with both preferences included, and then change the importance of the preferences depending on the ensemble, with zero importance for preferences that should not have any influence (this is more convenient than writing similar code four times, see the accompanying code for details). For the ensembles with each preference alone, we choose to study the effect of the preferences when they have high importance, to see their effects most clearly. For the ensemble where the two preferences act simultaneously, we choose to study the case where they have equal importance, and we set it to the highest possible (which in this case is medium importance because they have to share the importance, see details about the importance argument in the function help page).

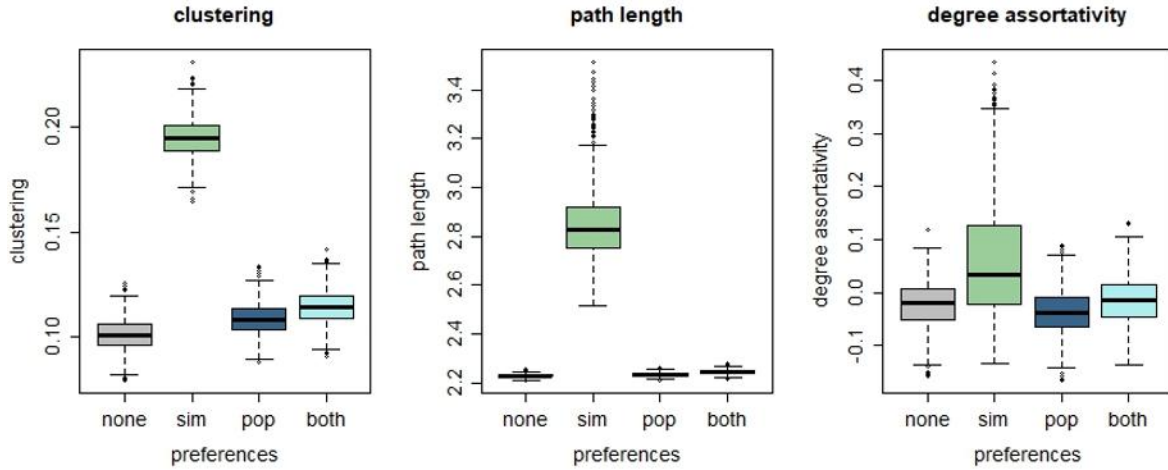


Figure 4. Measuring structural aspects of networks based on sex similarity and body size popularity trait preferences. The networks were generated and measured with the *traitnetmetrics* function. The figure shows three network metrics measured on four network ensembles: none = no preferences, sim = sex similarity preference, pop = size popularity preference, both = both preferences together.

We find that the sex similarity preference leads to networks with increased clustering and path length, and somewhat increased degree assortativity, compared to the networks with no preferences (Fig. 4). In contrast, the size popularity preference leads to networks with metrics that are similar to those of the networks with no preferences. Thus, the sex similarity preference has a clear effect on the measured structural aspects of network structure, whereas the size popularity preference has a limited effect.

Interestingly, the networks with both preferences are much more similar to those based on the size popularity preference than those based on the sex similarity preference in terms of the metrics - despite the fact that we set the two preferences to have equal importance when they were acting together. This can be explained by the finding (shown in Brask et al. 2026) that the effect of trait preferences on network structure tends to increase non-linearly with preference importance, and some preferences therefore can have relatively little effect at medium importance compared to at high importance.

In our visual exploration above we saw that networks based on the two preferences with equal importance (Fig. 3c) showed a clear pattern in terms of individuals of the same sex being closer in the network. This suggests that while the sex similarity preference with medium importance has limited effects on the three network metrics we measured here, it still has clear effects on the network, which could be quantified by measuring additional metrics, such as modularity or assortment by sex. Likewise, the size popularity preference may also have a stronger effect on other metrics, such as centralization (Fig. 3e). See the accompanying code for an example of quantifying network measures that are currently not included in okaapi.

Furthermore, it is known that the effect of trait preferences depends not only on the preference type but also on the trait value distribution (Brask et al. 2026). If our investigation concerned a specific real-world population, it could therefore be relevant to use trait values (e.g. sexes and body sizes) from the real population - if such were available - to understand the effect of the preferences on network structure in that specific population. See the code for an example.

Finally, here we used an approach where we made four network ensembles and compared them. This approach makes it possible to make easily interpretable box plots (Fig. 4), but it only tells us about the effect of the preferences for the specific preference importance settings we used for each ensemble. An alternative is to generate network ensembles across the whole range of importance for each trait preference (separately and together) and look at the metrics as a function of importance (as in Brask et al. 2026). This requires more simulations, but gives a fuller picture as it shows the effect of the preferences across the full ranges of importance. Which of these approaches are most useful depends on the specific study.

Concerning our aim of gaining insights into the effect of the two preferences on network structure, we can conclude that sex similarity preference has clear effects on network structure when the preference is highly important. Furthermore, our visual exploration suggests that the network metrics we used may not have comprehensively quantified the structural effects, and more investigations are needed to investigate effects of the size popularity preference, and of the sex similarity preference at lower importance.

7. Conclusion

Here we have presented the R package *okaapi*, which provides tools for generating, visualising and quantifying networks that are based on trait preferences (or equivalent node effects).

In the future we will likely expand the functionalities of the package by including additional similarity and popularity preference types, trait types and/or network metrics.

The *okaapi* package fills a gap by providing flexible network simulation tools that focus on trait preferences - a key driver of real social networks which is of ubiquitous importance across species. We hope that the package can contribute to exciting new discoveries about networks and sociality.

Acknowledgements

J.B.B. received funding from the Carlsberg Foundation (CF20-0663) and the Novo Nordisk Foundation (NNF23OC0085517). D.D.M. and L.J.N.B. acknowledge funding from a European Research Council Consolidator Grant awarded to L.J.N.B. (FriendOrigins—864461). D.D.M. acknowledges funding from the Max Planck Society.

Author contributions

J.B.B. conceived the idea of the okaapi package and developed it. The examples of how to use the package were developed by all authors and implemented by J.B.B. The writing of the manuscript was led by J.B.B., and all authors contributed significantly to the manuscript and approved the final version.

Conflict of interest statement

The authors declare no conflicts of interest.

Supplementary material

The okaapi user guide can be found here:

<https://github.com/bohrbrask/okaapi-user-guide>

See Table 1 for an overview of where to find okaapi resources.

Code availability

The currently available version of the okaapi package is called okaapibeta. Despite the name, it is fully functioning. It will be updated to okaapi at a later stage. The okaapibeta package is available here:

<https://github.com/bohrbrask/okaapibeta>

The package can be installed directly from within R in the following way (see accompanying R codes for demonstration): first install the 'remotes' or 'devtools' package, and then run the following code:

```
install_github("bohrbrask/okaapibeta")
```

The R codes with examples and demonstration can be found here:

<https://github.com/bohrbrask/okaapi-code-examples>

References

Brask, J. B., & Brask, J. B. (2024). Evolution of cooperation in networks with well-connected cooperators. *Network Science*, 12(3), 305-320.

- Brask, J. B., Silk, M., & Weiss, M. N. (2025). An introduction to generative network models and their use in animal sociality research. *Animal Behaviour*, 123364.
- Brask, J. B., Koher, A., Croft, D. P., & Lehmann, S. (2026). Far-reaching consequences of trait preferences for animal social network structure and function. *Behavioral Ecology*, 37(1), araf132.
- Cantor, M., & Farine, D. R. (2018). Simple foraging rules in competitive environments can generate socially structured populations. *Ecology and evolution*, 8(10), 4978-4991.
- Cantor, M., Chimento, M., Smeele, S. Q., He, P., Papageorgiou, D., Aplin, L. M., & Farine, D. R. (2021). Social network architecture and the tempo of cumulative cultural evolution. *Proceedings of the Royal Society B*, 288(1946), 20203107.
- Csárdi, G. & Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.
- De Moor, D., Hart, J. D., Franks, D. W., Brent, L. J., Silk, M. J., & Brask, J. B. (2025). Latent layers in social networks and their implications for comparative analyses. *Behavioral Ecology*, 36(6), araf113.
- Evans, J. C., Silk, M. J., Boogert, N. J., & Hodgson, D. J. (2020). Infected or informed? Social structure and the simultaneous transmission of information and infectious disease. *Oikos*, 129(9), 1271-1288.
- Farine, D. R., & Strandburg-Peshkin, A. (2015). Estimating uncertainty and reliability of social network data using Bayesian inference. *Royal Society open science*, 2(9), 150367.
- Franks, D. W., Ruxton, G. D., & James, R. (2010). Sampling animal association networks with the gambit of the group. *Behavioral ecology and sociobiology*, 64, 493-503.
- Ilany, A., & Akçay, E. (2016). Social inheritance can explain the structure of animal social networks. *Nature communications*, 7(1), 1-10.
- Mallon, D., Kümpel, N., Quinn, A., Shurter, S., Lukas, J., Hart, J.A., Mapilanga, J., Beyers, R. & Maisels, F. 2015. *Okapia johnstoni*. The IUCN Red List of Threatened Species 2015: e.T15188A51140517. <https://dx.doi.org/10.2305/IUCN.UK.2015-4.RLTS.T15188A51140517.en>.
- Newman, M. (2018). *Networks*. Oxford university press.
- Romano, V., Shen, M., Pansanel, J., MacIntosh, A. J., & Sueur, C. (2018). Social transmission in networks: global efficiency peaks with intermediate levels of modularity. *Behavioral ecology and sociobiology*, 72, 1-10.
- Ross, C. T., McElreath, R., & Redhead, D. (2024). Modelling animal network data in R using STRAND. *Journal of Animal Ecology*, 93(3), 254-266.
- Sah, P., Leu, S. T., Cross, P. C., Hudson, P. J., & Bansal, S. (2017). Unraveling the disease consequences and mechanisms of modular structure in animal social networks. *Proceedings of the National Academy of Sciences*, 114(16), 4165-4170.

Silk, M. J., Jackson, A. L., Croft, D. P., Colhoun, K., & Bearhop, S. (2015). The consequences of unidentifiable individuals for the analysis of an animal social network. *Animal Behaviour*, 104, 1-11.

Silk, M. J., & Gimenez, O. (2023). Generation and applications of simulated datasets to integrate social network and demographic analyses. *Ecology and Evolution*, 13(5), e9871.

Weiss, M. N., Franks, D. W., Brent, L. J., Ellis, S., Silk, M. J., & Croft, D. P. (2021). Common datastream permutations of animal social network data are not appropriate for hypothesis testing using regression models. *Methods in Ecology and Evolution*, 12(2), 255-265.