



HAL
open science

RCaNmodel: An R package for Chance and Necessity modelling

Hilaire Drouineau, Benjamin Planque, Christian Mullon

► **To cite this version:**

Hilaire Drouineau, Benjamin Planque, Christian Mullon. RCaNmodel: An R package for Chance and Necessity modelling. *Journal of Open Source Software*, 2023, 8 (82), pp.4955. 10.21105/joss.04955 . hal-04067325

HAL Id: hal-04067325

<https://hal.science/hal-04067325v1>

Submitted on 13 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License




RCaNmodel: An R package for Chance and Necessity modelling


Hilaire Drouineau ¹, Benjamin Planque ², and Christian Mullon ³

1 INRAE, UR EABX, 50 avenue de Verdun, CEDEX, 33612 Cestas, France 2 Institute of Marine Research, Norway, P.O. Box 6606, 9296 Tromsø, Norway 3 IRD, UMR MARBEC, Avenue Jean Monnet, Sete, France  Corresponding author

DOI: [10.21105/joss.04955](https://doi.org/10.21105/joss.04955)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: Fabian Scheipl 

Reviewers:

- [@juanvillada](#)
- [@amoeba](#)
- [@fabian-s](#)

Submitted: 27 September 2022

Published: 03 February 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

RCaNmodel is an R package dedicated to the development and exploration of time-dynamic food-web models. The associated user-friendly JavaFX graphical user interface (GUI) RCaNconstructor is specifically developed to facilitate the co-construction of such models with non-modelers. Together, RCaNmodel and RCaNconstructor ensure the traceability and replicability of food-web model developments, and provide a fast and efficient solution to handle the complex mathematical steps that arise from dynamic linear inverse modelling and polytope sampling in high dimensions. The software includes standard graphical outputs for the exploration of food-webs dynamics and the detection of trophic controls, which are key questions in trophic ecology.

Statement of need

Food-web models are widely used to study the functioning of ecosystems and to explore the impact of different pressures such as fishing or environmental changes ([Addison et al., 2013](#); [Murphy et al., 2012](#)). Food-web models primarily focus on trophic exchanges, also called flows or fluxes, between prey and predators. Estimations of trophic flows and of other model parameters is a complex task because the number of field observations is generally far smaller than the number of estimates from model outputs. Food-web reconstructions have no single best solutions, but rather an infinite number of flow patterns that comply with observational data, and food-web models are said to be underdetermined ([Niquil et al., 2011](#)). Several strategies have been developed to tackle this issue. In Ecopath ([Polovina, 1984](#)), or in its dynamic extension called Ecosim ([Christensen & Walters, 2004](#); [Pauly et al., 2000](#)) one of the most largely used trophic food-web models ([Heymans et al., 2016](#)) the modeler is asked to provide additional information on species diet or on their vulnerabilities to turn the underdetermined into an overdetermined problem. This approach ignores the highly variable and stochastic nature of trophic interactions. Food-web models based on linear inverse modelling (LIM) have been proposed as an alternative approach ([Vézina & Platt, 1988](#)). LIM models rely on the same key assumptions as of any food-web models (e.g., mass conservation) and are constructed using constraints (i.e., linear equalities/inequalities) that specify which combinations of flows are possible or not. Mathematically, the set of constraints defines a polytope: a fraction of the space that includes all the possible solutions satisfying the constraints (i.e. all possible trophic food web states). The polytope can be sampled using appropriate Markov chain Monte Carlo (MCMC) algorithms ([Kones et al., 2006, 2009](#); [Van den Meersche et al., 2009](#)). Like Ecopath, LIM is a static model that describe food-webs at equilibrium, but unlike Ecosim, there is no direct dynamic extension of LIM models. Recently, Planque and Mullon ([Planque & Mullon, 2020](#)) have developed a time-dynamic food-web modelling approach analogous to LIM, based on a generic approach called “Chance

and Necessity" (CaN) modelling. CaN models account for Chance, i.e. the stochasticity of nature and for Necessity, i.e., the existence of physical (e.g. mass conservation) or ecological (e.g. inertia of populations, satiation of individuals) constraints that can separate food-web dynamics that are possible from those that are not. CaN models also use observations to constrain the reconstruction of past food-web dynamics (Figure 1). As in LIM, building a trophic food-web CaN model starts with 1) the definition of the structure of the food web (who eats whom), followed by 2) the specification of the constraints (what is possible and what is not) and 3) the translation of these into the mathematical expression of a polytope. The validity of the resulting model is then tested by 4) checking that the polytope is bounded (i.e. infinite values are excluded) and not empty (i.e. some solutions exist). The following step is 5) to sample solutions within the polytope (i.e. time series of biomass and flows that satisfy all-time series of constraints). The complexity of steps 3, 4, and 5 is much greater in CaN than in LIM because the dynamic modelling in trophic food-web CaN entails much larger polytope dimensions (often >100 or even >1000). CaN model outputs/samples are then 6) analysed, often in a graphical manner, to reveal dynamical interactions between food-web components. Because outputs from food-web models are relevant outside the modelling community, e.g., for managers and stakeholders, there is a need for a modelling process - from model building to results interpretation - that can be conducted in a participatory manner. RCaNmodel and RCaNconstructor provide a mathematically efficient solution to steps 3, 4, and 5 and an interactive/intuitive interface to support steps 1, 2, and 6 in a participatory framework.

Methods

RCaNmodel and RCaNconstructor support the full modelling process from the specification of the model to the exploration of the outputs (Figure 2), while providing user-friendly solutions to each technical challenge:

- To specify the model, the user provides several tables that define the compartments (i.e., trophic groups), their ecological properties (species specific input parameters), the flows among the compartments (i.e., trophic interactions), observational time-series, and user-defined explicit constraints. Tables can be provided either via an RCaN file that consists of several spreadsheets in the .xlsx open format (see supplementary material for an example), or directly in the form of R data frames. The RCaN file format facilitates data entry for non-expert users, and exchange of input files between users. RCaN files can be constructed using the RCaNconstructor GUI, which ensures internal consistency between the data tables. Alternatively, data tables can be provided in R dataframe format to facilitate the use of external databases or version control through the use of plain text files.
- Based on the input data provided in the tables, the function *buildCaN* constructs the matrices that define the polytope (see Planque & Mullon (2020) for the underlying mathematics). The use of symengine (Ma et al., 2020), a library for symbolic manipulation, facilitates the automatic translation of user-specified literal constraints (see supplementary material for details on the syntax) into matrices.
- The function *checkPolytopeStatus* checks if the polytope is bounded and not empty, two conditions that are necessary for the food-web model to be valid. If the polytope is empty, the function *findingIncompatibleConstr* can be used to identify the minimum set of constraints that need to be relaxed to make the polytope feasible (inspired from <http://web.mit.edu/lpsolve/doc/Infeasible.htm>). Functions *getBoundParam* and *getAllBoundsParam* return the possible minimum and maximum values of all dimensions of the polytope, i.e., of all flows at all time-steps and of all biomasses at time-step 1. RCaNmodel uses linear programming tools to perform these tasks, using the package *lpSolveAPI* (Konis & Schwendinger, 2020).
- The function *sampleCaN* samples valid food-web trajectories using a Gibbs sampling algorithm that has proved to be especially efficient to achieve a uniform sampling in high dimensions polytopes (Andersen & Diaconis, 2007; Laddha & Vempala, 2020). The

function returns an object of class `mcmc.list` that can be manipulated/explored using the package `coda` (Plummer et al., 2010), e.g. to check the convergence of the MCMC algorithm.

- Finally, several graphical functions are available to explore the results per trophic groups and time periods, including diagrams describing the dynamics of flows and biomass, relationships among species and trophic controls (Figure 3).

All these steps can be done either directly through R scripts or using the `RCaNconstructor` GUI. An example using R commands is presented in the supplementary material. It should be noted that most of the functions to check the status or to sample the polytope can be used with other packages (e.g. LIM modelling package). Finally, the chance and necessity modelling approach is applicable to a wide range of problems beyond food-web models, and `RCaNmodel` is currently being generalised to other kind of models that rely on similar principles.

Figures

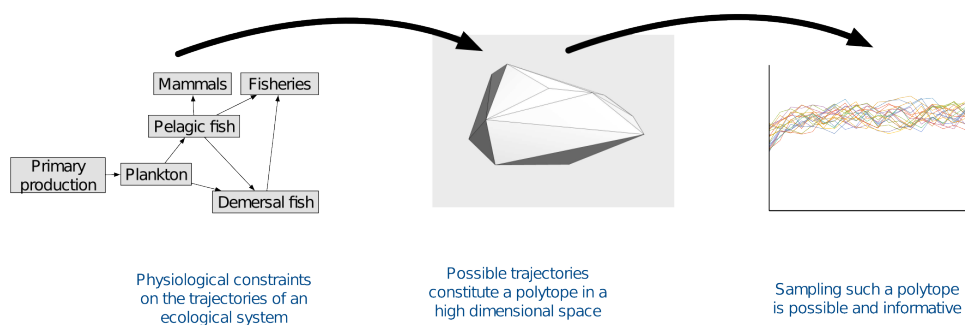


Figure 1: The CaN approach applied to a trophic food-web: physical and ecological constraints are used to delineate what food-web dynamics are possible or not. These linear equalities and inequalities define a polytope, i.e. a subspace that include all possible time-trajectories of the food-web and that can be sampled using appropriate MCMC algorithms.

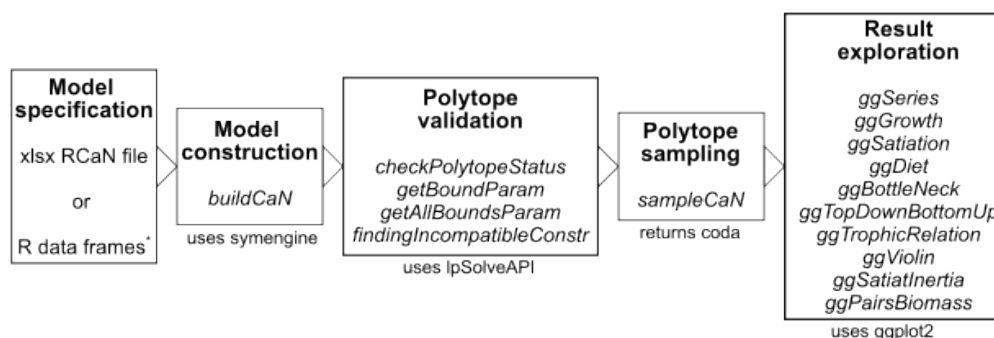


Figure 2: Overview of the modelling process to develop and explore a CaN trophic food web model with `RCaNmodel`. All the steps can be achieved using either an R script or the `RCaNconstructor` GUI, except the step denoted with a *. Italicised texts denote `RCaNmodel` functions. The boxes stand for the main steps, under which is mentioned their links with other R libraries.

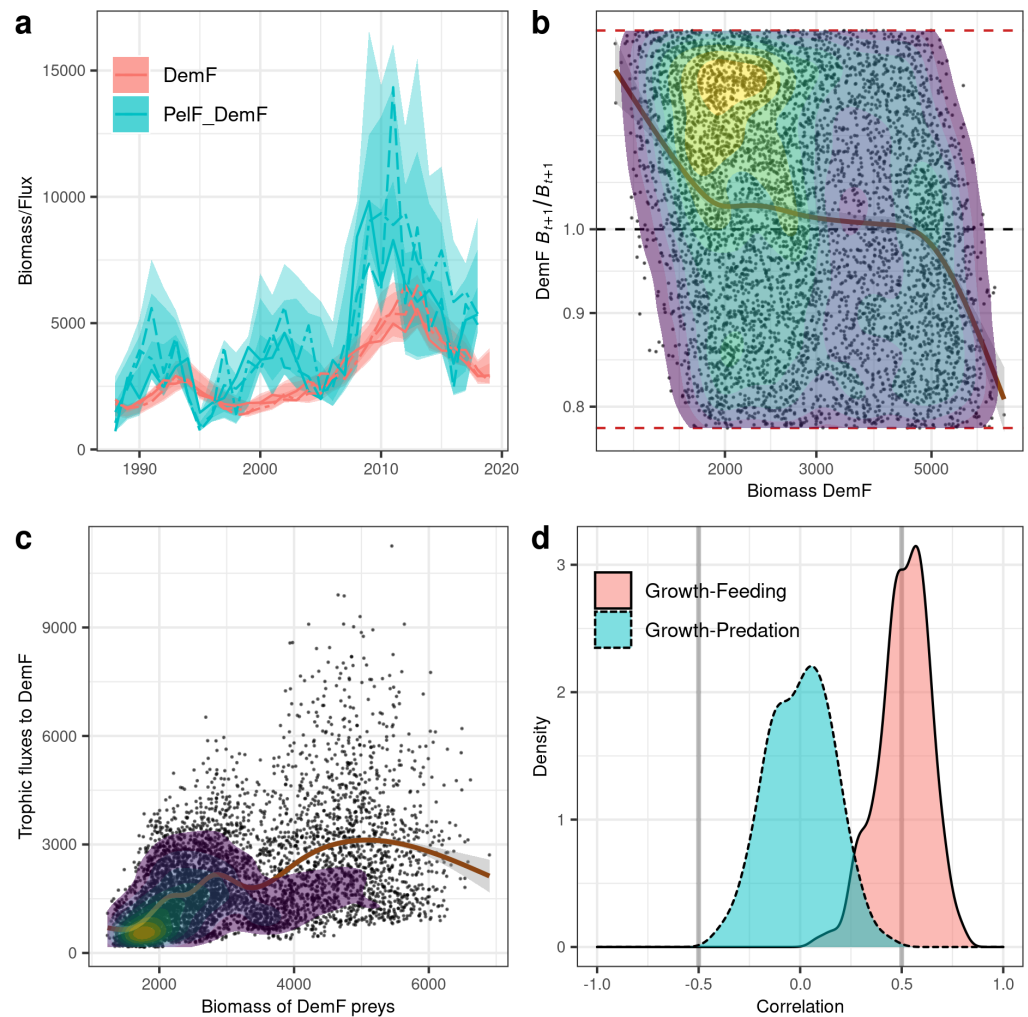


Figure 3: Example of diagnostic diagrams provided by *RCaNmodel*, based on the example provided as a supplementary material. Panel a shows the time variations of the biomass of a trophic group (DemF = demersal fishes) and the trophic flow from PelF (Pelagic fishes) to DemF (the lines correspond to 3 random trajectories while the ribbons corresponds to quantiles 50%, 97.5% and 100% among trajectories). Panel b illustrates how the growth rate of DemF (y-axis) decreases with the biomass of the stock (x-axis) among the different years and MCMC iterations, suggesting a limitation by resources. Panel c illustrates that the trophic flow towards DemF is correlated to the amount of available preys, suggesting a bottom-up control of DemF. Panel d shows that the growth of DemF (MM) is positively correlated to its feeding, suggesting a strong bottom-up control. On the other hand, the growth rate is not correlated to the predation by predators suggesting that there is no top-down control.

References

- Addison, P. F. E., Rumpff, L., Bau, S. S., Carey, J. M., Chee, Y. E., Jarrad, F. C., McBride, M. F., & Burgman, M. A. (2013). Practical solutions for making models indispensable in conservation decision-making. *Diversity and Distributions*, 19(5-6), 490–502. <https://doi.org/10.1111/ddi.12054>
- Andersen, H. C., & Diaconis, P. (2007). Hit and run as a unifying device. *Journal de La Société Française de Statistique*, 148(4), 5–28.
- Christensen, V., & Walters, C. J. (2004). Ecopath with Ecosim: Methods, capabilities and

- limitations. *Ecological Modelling*, 172(2), 109–139. <https://doi.org/10.1016/j.ecolmodel.2003.09.003>
- Heymans, J. J., Coll, M., Link, J. S., Mackinson, S., Steenbeek, J., Walters, C., & Christensen, V. (2016). Best practice in Ecopath with Ecosim food-web models for ecosystem-based management. *Ecological Modelling*, 331, 173–184. <https://doi.org/10.1016/j.ecolmodel.2015.12.007>
- Kones, J. K., Soetaert, K., Oevelen, D. van, & Owino, J. O. (2009). Are network indices robust indicators of food web functioning? A Monte Carlo approach. *Ecological Modelling*, 220(3), 370–382. <https://doi.org/10.1016/j.ecolmodel.2008.10.012>
- Kones, J. K., Soetaert, K., Oevelen, D. van, Owino, J. O., & Mavuti, K. (2006). Gaining insight into food webs reconstructed by the inverse method. *Journal of Marine Systems*, 60(1-2), 153–166. <https://doi.org/10.1016/j.jmarsys.2005.12.002>
- Konis, K., & Schwendinger, F. (2020). *IpSolveAPI: R interface to IpSolve* [Manual]. <https://CRAN.R-project.org/package=IpSolveAPI>
- Laddha, A., & Vempala, S. (2020). Convergence of Gibbs Sampling: Coordinate Hit-and-Run Mixes Fast. *arXiv:2009.11338 [Cs]*. <https://doi.org/10.48550/arXiv.2009.11338>
- Ma, J., Fernando, I., & Chen, X. (2020). *SymEngine: Interface to the SymEngine library* [Manual]. <https://github.com/symengine/symengine.R>
- Murphy, E. J., Cavanagh, R. D., Hofmann, E. E., Hill, S. L., Constable, A. J., Costa, D. P., Pinkerton, M. H., Johnston, N. M., Trathan, P. N., Klinck, J. M., Wolf-Gladrow, D. A., Daly, K. L., Maury, O., & Doney, S. C. (2012). Developing integrated models of Southern Ocean food webs: Including ecological complexity, accounting for uncertainty and the importance of scale. *Progress in Oceanography*, 102, 74–92. <https://doi.org/10.1016/j.pcean.2012.03.006>
- Niquil, N., Saint-Béat, B., Johnson, G. A., Soetaert, K., Van Oevelen, D., Bacher, C., & Vézina, A. F. (2011). Inverse modeling in modern ecology and application to coastal ecosystems. In E. Wolanski & D. McLusky (Eds.), *Treatise on estuarine and coastal science*. (pp. 115–133). Academic Press. <https://doi.org/10.1016/B978-0-12-374711-2.00906-2>
- Pauly, D., Christensen, V., & Walters, C. (2000). Ecopath, Ecosim, and Ecospace as tools for evaluating ecosystem impact of fisheries. *ICES Journal of Marine Science*, 57(3), 697–706. <https://doi.org/10.1006/jmsc.2000.0726>
- Planque, B., & Mullon, C. (2020). Modelling chance and necessity in natural systems. *ICES Journal of Marine Science*, 77(4), 1573–1588. <https://doi.org/10.1093/icesjms/fsz173>
- Plummer, M., Best, N., Cowles, K., & Vines, K. (2010). *Coda: Output analysis and diagnostics for MCMC*. <http://CRAN.R-project.org/package=coda>
- Polovina, J. J. (1984). An overview of the Ecopath model. *Fishbyte*, 2, 5–7.
- Van den Meersche, K., Soetaert, K., & Van Oevelen, D. (2009). Xsample(): An R function for sampling linear inverse problems. *Journal of Statistical Software*, 30, 1–15.
- Vézina, A. F., & Platt, T. (1988). Food web dynamics in the ocean. 1. Best-estimates of flow networks using inverse methods. *Marine Ecology Progress Series. Oldendorf*, 42(3), 269–287. <https://doi.org/10.3354/MEPS042269>