

1 **RCAN : A SOFTWARE FOR CHANCE AND NECESSITY**
2 **MODELLING**

3 HILAIRE DROUINEAU, BENJAMIN PLANQUE, AND CHRISTIAN MULLON

ABSTRACT. Uncertainty is a key challenge in modelling marine systems and a source of misunderstandings between scientists and other approach. Recently, a Chance and Necessity approach has been proposed to address this issue in trophic network modelling. Instead of seeking for a precise description of the system functioning and trajectory, CaN approach (a) focuses on the few certain physical and physiological laws and (b) samples all trajectories that can satisfy these laws, while being compatible with observations that have been collected about the system. This type of approach is thought to facilitate discussion among actors since it is based on information shared by most actors, it does not presume any knowledge of modelling practices and avoids knockout arguments based on scientific authority.

However, to be a good discussion support, user-friendly software are required to easily implement such models. In this paper, we present a software allowing to quickly develop this approach about a marine ecosystem that has been observed for several years. RCaN is a library of R commands. It implements efficient algorithms allowing to sample polytopes in spaces with a very large number of dimensions. RCaNconstructor is a graphical interface to RCaN.

We recall the main principles of CaN modelling, we present the software that has been developed and we give all the details of its use to the analyze of the trophic marine system of the Barents sea.

1. INTRODUCTION

Modelling complex ecological systems is often a trade-off between the accurate representation of the many components and interactions within an ecosystem and the reduction of ecosystem processes to few and important elements. Rather than attempting to model multiple ecological processes in details, CaN modelling focuses on exploring possible ecological dynamics given a set of constraints. The principles of CaN modelling and the corresponding mathematical formulation have been outlined in [1]. The general idea is that it is easier for various actors (managers, scientists, stakeholders) to agree about ecological constraints, which separate what is possible from what isn't, than to agree on the mechanistic formulation of detailed ecological processes. This is particularly true when modelling complex ecological systems for which available information/observations are limited. This approach is called "Chance and Necessity" modelling (CaN): chance refer to the indeterminacy in ecological processes while necessity reflects the constraints that delineate what is possible from what isn't.

While the CaN principles can be applied to a wide range of ecological problems, the first application was specifically designed to explore the temporal dynamics of marine systems [1]. This has been motivated by the shift from conventional single-species fisheries management towards ecosystem-based fishery management (EBFM, see for example [2, 3]). This shift implies upscaling of the traditional population scale management into a complex set of biotic (e.g. trophic food interactions) and abiotic sets of interactions (e.g. interactions with physical habitats). This has led to the development of complex ecological models, but which ability to help management is questioned by many actors who may have limited trust in such models [4].

28 In this context, there is a call for transparency in the science that supports man-
29 agement, and participatory modelling and assessment have been proposed a a way
30 to rebuild mutual trust. CaN is proposed as one tool to support transparency and
31 participatory modelling by providing a simplified food-web modelling framework, in
32 which the ideas of uncertainty and ecological limits are central. Assumptions about
33 knowns and unknowns and about possible and impossible processes are presented
34 explicitly and all model constituents can be described in plain language (little or
35 no modelling jargon).

36 By contributing to the transfer of biomass, energy and pollutants in ecosystems,
37 trophic interactions play a major role in ecosystem functioning. Trophic food web
38 models have proved to be key tools for the quantification of these fluxes, they
39 provide a holistic view of ecosystem functioning, and are increasingly used to assess
40 ecosystem ecological status and to explore the effect of anthropogenic pressures
41 such as fisheries [5]. Food web models, mostly Ecopath with Ecosim (EwE) [6, 7],
42 have played an important role in the implementation of the EBFM. CaN modelling
43 further contributes to this effort by providing a modelling framework for marine
44 food-webs that primarily relies on ecological constraints which can be agreed upon
45 by diverse actors.

46 The principles of CaN modelling have been outlined in [1]. The main technical
47 difficulty: sampling a polytope in a highly dimensional space have been tackled
48 inside the R framework [8]. Practical results have been obtained [1].

49 Until now, there existed no software to easily perform CaN modelling. The first
50 models were constructed in an *ad hoc* fashion by assembling and structuring relevant
51 data and developing tailor-made programs in Mathematica or R to run the model.

52 This made it difficult to communicate and replicate CaN models. It also restrained
53 the access of the model to non-modellers. Making CaN modelling transparent, repli-
54 cable, and performant in a participatory modelling framework requires that model
55 construction, sampling and output visualisation be easily accessible by modellers
56 and non-modellers alike, and that the structure of CaN models be standardised. A
57 graphical user interface (GUI), which provides an easy entry level for users is also a
58 highly desirable feature for a modelling approach targeting non-modellers. In this
59 contribution, we present a software for the implementation of CaN modelling in R,
60 called RCaN, and an associated GUI in Java, called RCaNconstructor. RCaN and
61 RCaNconstructor can be used jointly or separately. Their combined use allow users
62 to easily construct, document, sample and interpret the results of CaN food-web
63 models. The relationship between RCaN and RCaNconstructor is shown in figure
64 1. The main features of RCaN and RCaNconstructor include:

- 65 (1) a user-friendly software that can handle the complex numerical and math-
66 ematical parts of the modelling. In such way, users can then focus on the
67 discussion about their knowledge of the ecological system and how to best
68 structure a model that fits their needs,
- 69 (2) a file and data management system that ensures consistency between the
70 different elements of a CaN model. For example, it must be guaranteed that
71 each link connects existing components or that each constraint points to
72 components, links or data series that are adequately specified in the model,
- 73 (3) a standardised file format to fully specify a CaN model,

- (4) Importation of observation data-files often result in the repetition of many procedures: selecting an external file, selecting year range and observations in the file, etc., with the the risk of incoherencies, forgettings, etc.
- (5) a set of R functions that allow experimented programmers to construct, manipulate and visualise the outputs of CaN models in a flexible manner,
- (6) standardised graphical outputs of the model results, which promote discussions among modellers and other actors and ease the interpretation of model results,
- (7) a system for registering meta-information about the model, such as model version, authors, data sources, assumptions, and so on. This ensures transparency and reproducibility of modelling experiments.

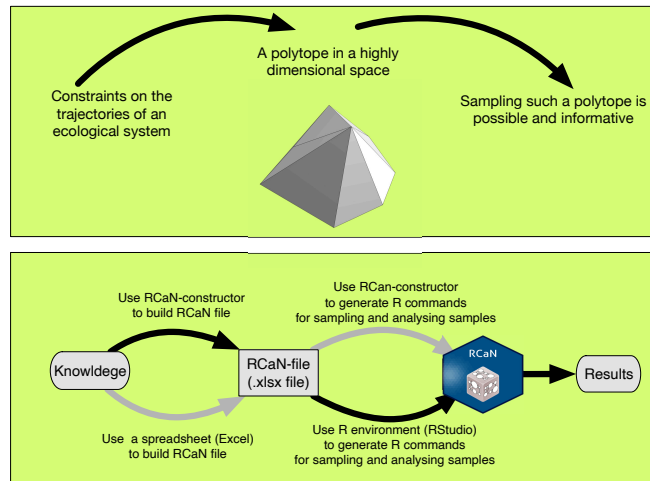


FIGURE 1. Top: Principles of CaN modelling. Bottom: RCaN and RCANconstructor. The usual ways of using RCAN, with a common element, a RCaN-file.

In section 2 we summarise the elements of the CaN modelling approach. In section 3, we present the format of an RCaN file. In section 4 we present the R library RCaN which is designed to construct, sample and produce graphical outputs

88 of CaN food-web models. In section 5 we introduce RCaNconstructor, the graphical
 89 user interface that facilitates the co-construction of CaN models and the inclusion
 90 of meta-information. In section 6, we briefly describe the Barents Sea trophic
 91 network which we use as a case study example. Finally, we discuss the possible
 92 future developments of the RCaN and RCaNconstructor softwares.

93 2. PRINCIPLES OF THE CAN APPROACH

94 In CaN, food-web dynamics is defined by time-trajectories of biomass and fluxes
 95 between trophic groups. The biomass of all trophic groups and the fluxes between
 96 them can usually not be precisely observed. Therefore, the food-web dynamics is
 97 indeterminate, and a range of dynamics is possible. However, not everything is
 98 possible because some ecological or physiological constraints are known to bound
 99 the temporal evolution of ecological system, and/or because some observations can
 100 inform about the past dynamics of some components of the food-web. These eco-
 101 physiological constraints and observations can be translated into mathematical con-
 102 straints that need to be satisfied before a food-web trajectory can be considered
 103 possible. The ensemble of these mathematical constraints constitutes the “neces-
 104 sity” in a CaN model. It is by focusing on constraints that can be easily expressed
 105 and communicated between actors and by explicitly acknowledging indeterminacy
 106 in natural systems that CaN modelling facilitate exchanges among actors to support
 107 resource management.

108 **2.1. Network structure.** In CaN modelling, the food-web is defined as a net-
 109 work. In this network, each vertex/node is called a component; it corresponds to a
 110 species, or more generally a trophospecies, i.e. a group or sub-group of species that
 111 share common prey and predators. Components may be within the model domain;

112 then their biomass trajectories are explicitly considered. They also may be outside
113 the model domain, then we have not knowledge of their biomass trajectories. An
114 edge/link corresponds to a flux of biomass from/to comp. Most fluxes are trophic:
115 from a prey to a predator. Some fluxes are non-trophic: they correspond to im-
116 port/export of biomass from adjacent regions or to removal of biomass through
117 fishing.

118 **2.2. Components characteristics.** Component inside the system are character-
119 ized by the following biological properties. **Satiation** reflects the maximum con-
120 sumption rate per unit biomass of the predator. **Inertia** expresses that variations
121 in biomass from one year to the next are bounded between a maximum growth
122 rate and a maximum mortality rate. **Digestibility** is a correction factor that ac-
123 counts for variations in energy content between prey. **Assimilation efficiency**
124 expresses the proportion of the biomass ingested by a predator that is effectively
125 assimilated; the product of the potential assimilation efficiency by the digestibility
126 correction factor is the absorption efficiency (the proportion of prey biomass di-
127 gested and absorbed). **Refuge biomass** is the absolute minimum biomass that a
128 species group can reach. **Other losses** is a mortality coefficient that account for
129 losses, i.e. metabolic losses and other mortality, not explicitly accounted for in the
130 model.

131 Components outside the model domain do not require input parameter except
132 when they constitute a prey, in which case digestibility should be provided.

133 **2.3. Observations.** The CaN modelling approach emphasizes the past observa-
134 tions of the dynamics of the food-web. Observations can be: catch or landings
135 data, biological samples of specific populations, stomach samples or other sources

of diet information, survey abundance estimates, outputs from stock assessment models, estimated ranges of biomass, and so on.

2.4. Constraints. Constraints that limit the dynamics of the food-web are at the heart of CaN modelling and they are the way by which most of the knowledge and data enter a CaN model. Some constraints are implicitly incorporated into CaN while others can be explicitly formulated.

A few implicit constraints are standard for all CaN models. They are related to the above components characteristics. Constraints are as follows: (a) **there is conservation of energy during trophic processes; it relates trophic flux and biomasses: for a given compartment, assimilated inflows must compensate outflows and maintenance losses that are proportional to biomass** ; resulting biomass of compartments are positive and above refuge biomass, (b) trophic fluxes between compartments are positive, (c) the sum of fluxes into a compartment is limited according to its biomass (satiation), (d) the proportional changes in biomass of a compartment are bounded (inertia).

Additional explicit constraints can be provided. These are written in the form of inequalities or equalities. The left side of the (in)equality must contains a reference to one or several compartments or fluxes. The right side of the (in)equality can contain fixed values, compartments, fluxes, data time series.

Roughly speaking, there are two kinds of additional explicit constraints. Some constraints are the expression of a knowledge about system functioning and relate the biomass of a components to its incoming or outgoing fluxes. Some other constraints relate the possible values of a biomass or a flux to observations. These last constraints constitute the way to deal with uncertainties in CaN modelling.

160 **2.5. Trajectories.** A food-web trajectory is defined by the ensemble of biomass
 161 fluxes at each time-step and by the initial food-web configuration, i.e. the initial
 162 biomass in each component. Each trajectory can be expressed as a single point in a
 163 multi-dimensional space of dimension $P = nF \times nt + nB_0$, with nF the number of
 164 fluxes, nt the number of time steps and nB_0 the number of initial biomasses. This
 165 dimension can be very high: a simple ecosystem has about 10 components and 20
 166 types of fluxes; following it during 15 time steps results in $P = 20 \times 15 + 10 = 310$.

167 All constraints are linear equalities or inequalities. They can be expressed as bear-
 168 ing on food-web trajectories. As constraints are linear, the set of the trajectories
 169 that satisfy all constraints has the shape of a convex polytope in a high dimensional
 170 space.

171 **2.6. Sampling trajectories.** The main tool of a CaN modelling approach is the
 172 sampling of possible trajectories given (1) the food-web structure, (2) the species
 173 biological properties, (3) available observations and (4) constraints. The collection
 174 of sampled trajectories expresses the indeterminacy in the system.

175 Sampling food-web trajectories is achieved by sampling this high-dimension poly-
 176 tope using dedicated sampling algorithms.

177 3. RCAN FILE

178 In RCAN, all input information is provided in a simple excel spreadsheet. In
 179 sheet **Components**, first column corresponds to the name of a component. The
 180 second column indicates whether the component is inside or outside the trophic
 181 network. Other columns correspond to components characteristics that have been
 182 presented above.

183 In sheet **Trophic fluxes**, each flux is characterized by an Id (first column) and
 184 links a source component (column From) to a sink component (column To) and by
 185 its type, trophic or non trophic (last column); in a trophic flux, digestibility and
 186 assimilation should be applied to the flux.

187 In sheet **Observations**, first column gives the identifier of time steps (most of the
 188 time years). Other columns correspond to the extracted time series of observations
 189 that can be used to constrain the system (see 2.3).

190 In sheet **Constraints**, rows correspond to the constraints on biomasses, fluxes
 191 and time series. Each constraint has an Id (first column), a formula defining the
 192 linear constraints involving fluxes, time series and biomass in components (second
 193 column), a time period of validity (third column) and a flag indicating whether the
 194 constraint should be used or not (fourth column).

195 What is important to notice is that the contents of these sheets are interrelated
 196 and that this results in some coherency rules.

197 4. RCAN

198 4.1. **Introduction.** RCaN is an R package aiming at building and running CaN
 199 trophic food web model. As explained in previous parts, the mathematical struc-
 200 ture to analyse is a set of analytical linear equalities and inequalities. A complete
 201 mathematical exposition is given in [1].

202 The rationale is that: (a) constraints related to elementary biological processes or
 203 to past observations result in linear equalities or inequalities involving the dynamics
 204 of the system, that is its trajectories, (b) these constraints define a set in a highly
 205 dimensional space, this set has the structure of a polytope, (c) a lot can learnt
 206 about the system by sampling this polytope.

207 Implementing the whole process implies (a) to specify entities of the system, (b)
 208 to express constraints involving entities and their trajectories, (c) transform set of
 209 constraints in a polytope of trajectories, (d) analyse elementary properties of the
 210 polytope (non-emptiness, bounds), (e) sampling the polytope, (f) analyse sample.

211 Next paragraphs are about technical features of the implementation of these
 212 different steps in the R language.

213 4.2. The different steps to use RCaN.

214 4.2.1. *Build a polytope from RCaN-file.* Once the RCaN-file has been built, a first
 215 step is to organize all the information and to build the polytope describing the
 216 considered model. More precisely, the mass conservation of biomasses and con-
 217 straints can be summarized in matrix notation, and all these matrices should be
 218 automatically generated whatever are the constraints and parameters used by the
 219 modeller.

220 Constraints can be summarized in a matrix form as $A \cdot x \leq b$ (inequality
 221 constraints) and $C \cdot x = v$ (equality constraints) with x a solution (e.g. a trajectory
 222 defined by initial biomasses per component and fluxes at each time step), A and
 223 C two matrices with as many columns as parameters in the model (fluxes per time
 224 step and initial biomass per component) and a row per constraint, and vector v
 225 and b the bounds of the inequalities.

226 Using the constraints formula and the symbolic links among variables, RCaN
 227 automatically build matrices C and A and vectors b and v for each active constraint.
 228 This is done by a simple call to the function **build_CaNmod** which reads all
 229 the information from the RCaN-file and returns a single object that represents all
 230 the trophic network representation and especially, all the corresponding polytope

matrices. For example, in the Barents Sea case study (section 6), the function automatically constructs the matrix A which has 2593 rows and 775 columns, and the matrix C which has 54 rows (and by definition, the same number of columns).

4.2.2. Checking polytope characteristics. At first try, the specification of model often leads to empty polytope (i.e. a model with no solution that satisfy all the constraints) or unbounded polytope (i.e. a model in which the specified constraints are not sufficient to bound a given parameter that can vary subsequently to infinity). Before running the model, it is wise to check the status of the polytope (the model won't run with an empty polytope while results are generally not relevant with unbounded polytope). The package provides different functions to check the properties of the polytope, of which the two most important are **checkPolytopeStatusCaNmod**, which checks whether the polytope is closed and non empty, and **getAllBoundsParamCaNmod**, which estimates the bounds for all parameters. If the model is empty, the package provides a function **findingIncompatibleConstraintCaNmod**, which indicates to the user which constraints raise problems.

4.2.3. Sampling polytope. Once all the previous steps are completed, the aim of this step is to achieve a uniform sampling within the convex polytope (if it is not empty). Achieving a uniform sampling is not straightforward since the a CaN polytope has a very high number of dimensions: the number of time steps times the number of internal components (corresponding to initial biomasses), plus the number of fluxes. This is the role of the function **fitmyCaNmod**, which carry out a uniform sampling within the polytope and returns results in the standard **coda** format of a **mcmc.list** [9]. Based on a Monte-Carlo sampler, the functions allows running several chains in parallel.

255 4.2.4. *Diagnostics about sampling.* One sampling is done, it is necessary to check
256 whether the uniform sampling was successful. Since the result of **fitCaNmod** is a
257 **mcmc.list** object, we can use all the diagnostics tools provided in the package **coda**,
258 such **traceplot**, **summary** or **gelman.diag** and rubin tests or autocorrelograms.

259 4.2.5. *Graphical analysis of sampling.* Each iteration of the MCMC sampling pro-
260 vides a possible set of time series of biomasses and fluxes. RCaN provides several
261 function to explore these results. Among others, **ggResult** and **ggViolin** can be
262 used to plot the temporal distributions of any biomass and/or flux. It is also possible
263 to explore the diet of the species using the function **ggDiet**, to look at the rela-
264 tionships among species with functions **ggPairsBiomass** or **ggTrophicRelation**,
265 or to focus on the biomass variations of specific species with functions **ggGrowth**
266 and **ggSatiation**.

267 4.3. **Technical considerations.** The development of the packages raises several
268 challenges. The first challenge was to improve computation performance. For this
269 purpose, RCaN is interfaced with **C++** using two R packages **Rcpp** [10] and
270 **RcppEigen** [11]. The second challenge was check the characteristics of the poly-
271 tope. This was achieved by using standard linear programming library: RCaN uses
272 **lp_solve** through the R package **lpSolveAPI** [12]). A challenge was also to trans-
273 late the content of a RCaN-file into matrices, and especially to allow user to specify
274 constraints in a simple language. This was made possible by using the symengine
275 R package [13], an interface to the symengine, a fast symbolic manipulation library
276 (the constraint syntax is presented in Supplementary Material).

277 Finally, the most challenging issue was to achieve the uniform sampling within
 278 the polytope given the dimensions of the polytope. To adress this problem, we im-
 279 plement two algorithms. Hit-and-run sampling have been proposed for a long-time
 280 to achieve uniform sampling of high-dimensional convex space [14][15]. Basically,
 281 hit-and-run sampling starts from a point in the polytope. Then at each iteration,
 282 a direction of the high dimensional space is randomly drawn and a new point is
 283 uniformly sampled between the two extreme bounds defined by the intersection
 284 with the envelop of the polytope. As such, at each iteration, the point is only
 285 moved in a random, but single direction, and consequently, when the sampler is
 286 "trapped" in a narrow zone of the polytope, it may take many simulation to pick
 287 the appropriate direction to exit the narrow zone resulting in autocorrelation. A
 288 variation of the hit-and-run sampler is referred as the Gibbs sampler or Coordinate
 289 Hit-and-Run [15][16]: instead of picking a direction and updating all coordinates
 290 in this direction, the Gibbs sampling picks randomly a dimension (i.e. a parameter
 291 of the model) and generate a new parameter value that satisfies the constraints.
 292 By doing so, it guarantees an efficient convergence [16]. By default, RCaN uses a
 293 Gibbs sample (but hit-and-run can be used), and its tuning is transparent for the
 294 user.

295

5. RCaNCONSTRUCTOR

296 Main goal of the RCaN constructor is to help the user to manage the RCaN-file.
 297 It insures the coherency of the RCaN-file (changing the characteristics of an object
 298 somewhere triggers the necessary changes elsewhere).

299 It allows to run RCaN commands. RCaNconstructor allows to keep a trace of
300 the meta information related to the scientific project of which the modelling is part
301 and of the data files that have provides the observation that constrain the model.

302 RCaNconstructor is a Java program using Javafx for building the graphical in-
303 terface and allowing the code to be imported and run on external computers; it
304 uses a Java library named **RCaller** for interfacing R and Java.

305 **5.1. Meta information.** RCaNconstructor allows to mobilize meta-infomation
306 about the model. In which context, for which purpose it is being carried out;
307 by whom? What is the provenance of datasets? All that information, of which
308 the recollection is often neglected, reveal themselves crucial, when it is the time to
309 communicate and justify results.

310 **5.2. Views.** RCaNconstructor allows the user to define the objets that are used
311 as inputs of the models. This is done with some dedicated views on the system,
312 its network structure, its components, observations and constraints. According to
313 the principles of RCaN modelling, views on the system are about: (a) the network
314 structure of the trophic system, (b) its components, (c) its fluxes, (d) observations
315 that have been selected on the system, (e) constraints that have been identified
316 about its dynamics.

317 From these different points of views, the user can add, edit or remove objects:
318 components, fluxes, observations, constraints. They are saved in a file which fulfill
319 the requirements of a RCaN-file.

320 **5.3. R interface.** The RCaNconstructor is interfaced to R with the RCaller library
321 [17]. From the RCaNconstructor menu, the user can interactively: (a) start a R
322 session and load RCaN library, (b) Build polytope (corresponds to the RCaNMod

323 command of RCaN), (c) sample polytope (corresponds to the fitMyMod command
 324 of RCaN), (d) get sampling diagnostics, (e) analyze sample.

325 6. EXAMPLE: THE BARENTS SEA FOOD-WEB

326 To illustrate CaN modelling using RCaN and RCaNconstructor, we use a sim-
 327 plified food-web of the Barents Sea as an example.

328 In what we present below, we have chosen to build the RCaN-file using RCaN-
 329 constructor and then to run RCaN commands to construct the polytope, sample
 330 it, and analyse the properties of the sample.

331 **6.1. Monitoring of the Barents Sea.** Monitoring of the Barents Ecosystem has
 332 been conducted for several decades and a collection of data series are reported
 333 regularly and compiled by the ICES working group on the integrated assessment
 334 of the Barents Sea (WGIBAR, [18]). These time series cover the period 1988-2019
 335 and include data on:

- 336 (1) landings for pelagic and demersal fish, krill, shrimps and marine mammals,
- 337 (2) survey-based biomass estimates of zooplankton, pelagic and demersal fish,
- 338 (3) satellite based estimate of Net Primary Production (NPP, from 1998),
- 339 (4) consumption estimates by Atlantic cod (*Gadus morhua*) of krill, shrimps,
- 340 capelin (*Mallotus villosus*), herring (*Clupea harengus*), polar cod (*Bore-*
 341 *ogadus saida*) and Atlantic cod.

342 In addition, there are estimates of the minimum and maximum plausible biomasses
 343 of benthos, marine mammals and birds which are used as limits for the whole
 344 time-period in the model ([19]).

345 **6.2. Meta Information.** Information about the model version, authors, funding
346 sources, data sources and pre-processing, assumptions and uncertainties are pro-
347 vided via the menu Meta-information/model description.

348 **6.3. Food-web Structure.** The model of the Barents Sea food web is composed of
349 11 components. Four of them are external (two plankton standing stocks from the
350 adjacent Norwegian Sea, primary producers and Fisheries), and seven are within the
351 model domain, and include plankton (Hzoo: herbivorous zooplankton, Ozoo: om-
352 nivorous zooplankton), benthos (bent), fishes (PelF: pelagic fishes, DemF: demersal
353 fishes) and top predators (MM: marine mammals, birds). These 11 components are
354 connected via 24 biomass fluxes among which 18 correspond to trophic relationships
355 and 6 to imports (plankton) or export (fisheries catches) of biomass.

356 With RCaNconstructor the food-web structure is specified by drawing the com-
357 ponents and fluxes (menu View/Network). Using this graphical interface (fig. 2,
358 top-left) it is possible interactively add, delete and edit components and fluxes.
359 The graphical positioning of the components of the network can also be modified
360 to ease the visual interpretation of the food-web structure.

361 **6.4. Input parameters.** The input parameters for individual components of the
362 Barents Sea food web (menu View/Components) are taken from [19]. These were
363 derived from the Metabolic Theory of Ecology, Life-history theory or previous mod-
364 elling exercises using for example Ecopath [20]. These are summarised in Table 1,
365 in the supplementary material.

366 **6.5. Observations.** Observations data can be be see as a table (Menu Views.
367 Submenu Observations). A button Button add an observation allows to create a
368 new observation, i. e., a new column in this table. This observation is coming

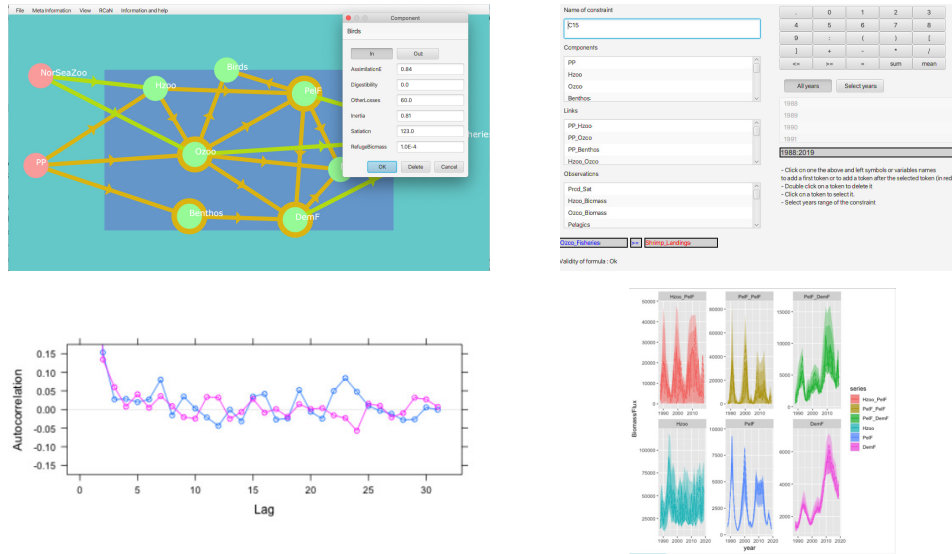


FIGURE 2. A selection of input/output windows from the RCaN-constructor applied to the Barents Sea food-web model. Top left: RCaNconstructor graphical interface to draw and specify the trophic network. Top right: the graphical interface used to specify the model constraints. Bottom left: sample diagnostic plot showing the autocorrelation of two chains of one flux. Bottom right: sampled trajectories (time-series) of selected fluxes and components.

369 from a file (a CSV file with a simple structure; first line: observations names; first
 370 column: years) that has to be specified in the interface and to and to a column
 371 that has to be selected in this file. For the Barents sea, we have used the following
 372 data sets: ...

373 **6.6. Constraints.** A RCaN model contains the following implicit constraints: sa-
 374 tiation, inertia, positive fluxes and refuge biomass. These are specified in the input
 375 parameters. Additional constraints can be added (Menu Views/Constraints/Add a
 376 constraint, fig. 2, top, right.). These explicit constraints reflect knowledge about
 377 the ecological system and can be used to link the model to available observations.
 378 In the Barents Sea case study, the additional constraints are defined as follows:

- 379 (1) The consumption of primary production by zooplankton and benthos is
380 constrained by net primary production estimates ($\pm 30\%$) for the period
381 when these estimates are available (1998 to 2019). For other years, this
382 sum is bounded by estimates of absolute maximum and minimum primary
383 production that are set to 1/2 million and 2 million tonnes respectively.
- 384 (2) Time-series of landings provide accurate information about the fluxes from
385 herbivorous and omnivorous zooplankton, pelagic and demersal fish, and
386 marine mammals to corresponding fisheries. Therefore, these fluxes can be
387 set to the values of the estimated landings.
- 388 (3) The annual estimates of biomass for herbivorous and omnivorous zooplank-
389 ton, and pelagic and demersal fish, derived from research surveys, can be
390 used to constrain the biomass of these groups.
- 391 (4) The estimates of minimum and maximum plausible biomasses for benthos,
392 marine mammals and birds can be used as limits for these groups for whole
393 time-period in the model.
- 394 (5) The food consumption by demersal fish can be bounded based on consump-
395 tion estimates by cod in the Barents Sea. These bounds are as follows:
- 396 (a) the consumption of demersal fish by demersal fish is greater than the
397 consumption of cod by cod,
- 398 (b) the consumption of pelagic fish by demersal fish is greater than the
399 consumption of capelin, herring and polar cod by cod;
- 400 (c) the consumption of omnivorous zooplankton by demersal fish is greater
401 than the consumption of krill and shrimp by cod

(d) the total consumption by demersal fish is not greater than twice the
the total consumption estimates by cod.

6.7. Polytope building. Once the components, input parameters, fluxes, observations and constraints are provided, the model is fully specified. This information is gathered in the corresponding RCaN file. The polytope can be constructed by running R commands of the RCaN library. This can be achieved within the RCaN-constructor environment or directly within the R environment. Both ways have their advantages and limitations. Using RcaNconstructor is intuitive and does not require prior knowledge of the R language. Using R provides a greater flexibility for users who are familiar with the R language. We provide below the commands in the R language.

We build the polytope with following R command :

```
library(RCaN)
CaNMod <- buildCaN("BarentsSeaReconstructions_20210227.xlsx")
```

In the Barents Sea case, the polytope has 847 dimensions (7 initial values - one for each component within the model domain - and 24 fluxes times 35 years). There are 2811 constraints (both implicit and explicit). Constructing this polytope on a laptop computer takes 80s (MacBook Pro Retina, 13-inch, Early 2015, 3.1 GHz Dual-Core Intel Core i7).

It is possible to check the status of the polytope (bounded, empty or unbounded) and to compute the bounds of the polytope in every of its 847 dimensions using the following commands (the time taken to obtain the results is indicated in round brackets):

```
checkPolytopeStatus(CaNMod) \\ (16s)
```

```
426 2 CanBounds <- getAllBoundsParam(CaNMod) \\ (2900s)
```

427 **6.8. Polytope sampling.** When the polytope is built, and if it is properly shaped
 428 (i.e. not empty or unbounded), it is then possible to sample it. Each sample
 429 corresponds to the trajectory of the food web and contains the initial biomass
 430 values for each component inside the model domain and the values of the fluxes
 431 at each time step. That is 847 values in the exemple of the Barents Sea model.
 432 Sampling the polytope is achieved with one RCaN command:

```
433 1 CanSample <- sampleCaN(CaNMod, N=100, thin=100, nchain=2, ncore=2, method  
434 = "gibbs") \\ (520s)
```

435 This command returns an R list object which contains the original model structure
 436 and two chains of 100 samples (i.e. 100 food-web trajectories). In total 10,000
 437 samples were explored in each chain, but only every 100th sample was retained.
 438 The polytope sampling algorithm used is the Gibbs sampler.

439 **6.9. Sample diagnostics.** An example of sampling diagnostics is shown in Figure
 440 1, bottom left. This shows the autocorrelation functions for the two chains, for the
 441 flux from primary production to benthos (PP_Benthos) in the year 2011. This is
 442 obtained with the following command:

```
443 1 acfplot(CaNSample$mcmc[, "PP_Benthos[2011]"], lag.max=30) \\ (<1s)
```

444 **6.10. Sample Graphical outputs.** An example of time-series outputs for 3 fluxes
 445 and 3 components is provided in Figure 1, bottom right. This is obtained with the
 446 following command:

```

447 1 library(ggplot2) \\
448 2 ggSeries(CaNSample, c("H zoo_PelF", "PelF_PelF", "PelF_DemF", "H zoo", "PelF"
449      , "DemF")) \\ (<1s)

```

450 In addition to the R function `ggSeries`, a collection of pre-defined plots are pro-
 451 vided. These can be used to quickly visualise the main model results. These include
 452 violin plots of biomass and fluxes (`ggViolin`), the diet fractions of the food-web
 453 components (`ggDiet`), the relationships between growth and biomass (`ggGrowth`),
 454 the correlations between components' biomass (`ggPairsBiomass`), the trophic func-
 455 tional relationships (`ggTrophicRelation`) or the relationships between ingestion and
 456 predator biomass (`ggSatiation`).

457 **6.11. Concluding.** Review initial objectives (RCaNmod, menu Meta Informa-
 458 tion). Communicate in terms of uncertainty.

459 7. DISCUSSION

460 The RCaN modelling approach has first been an attempt to directly address the
 461 issues of uncertainty and lack of knowledge in the modelling of exploited marine sys-
 462 tem. It has given several modelling experiments during which algorithms have been
 463 defined and improved. The R libraries for Bayesian modelling have revealed to be
 464 a powerful environment. On an other hand, the RCaN modelling approach implies
 465 different modelling protocols. The main issue is an adequate selection of constraints:
 466 practically, they must result in a non empty and bounded polytope. Contradiction
 467 between constraints allows to reconsider the scientific knowledge about a marine
 468 system. This is the goal of the RCaN constructor to make data management and

469 computing easier allowing then the necessary focus on the nature of scientific knowl-
470 edge about marine systems. Next step will consist in improving algorithms and in
471 deepening the specific methodology for RCaN.

472 REFERENCES

- 473 [1] Benjamin Planque and Christian Mullon. Modelling chance and necessity in
474 natural systems. *ICES Journal of Marine Science*, 2019.
- 475 [2] Stephen J Hall and B Mainprize. Towards ecosystem-based fisheries manage-
476 ment. *Fish and Fisheries*, 5(1):1–20, 2004.
- 477 [3] Elizabeth A Fulton, Anthony DM Smith, David C Smith, and Penelope John-
478 son. An integrated approach is needed for ecosystem based fisheries manage-
479 ment: insights from ecosystem-level management strategy evaluation. *PloS*
480 *one*, 9(1):e84242, 2014.
- 481 [4] Sigrid Lehuta, Raphaël Girardin, Stéphanie Mahévas, Morgane Travers-Trolet,
482 and Youen Vermard. Reconciling complex system models and fisheries advice:
483 Practical examples and leads. *Aquatic Living Resources*, 29(2):208, April 2016.
484 ISSN 0990-7440, 1765-2952. doi: 10.1051/alr/2016022.
- 485 [5] Éva E. Plagányi. Models for an ecosystem approach to fisheries. Report, Food
486 and Agriculture Organisation of the United Nations, 2007.
- 487 [6] Jeffrey J. Polovina. An overview of the Ecopath model. *Fishbyte*, 2:5–7, 1984.
- 488 [7] Johanna Jacomina Heymans, Marta Coll, Jason S. Link, Steven Mack-
489 inson, Jeroen Steenbeek, Carl Walters, and Villy Christensen. Best
490 practice in Ecopath with Ecosim food-web models for ecosystem-based
491 management. *Ecological Modelling*, 331:173–184, 2016. ISSN 0304-
492 3800. doi: <http://dx.doi.org/10.1016/j.ecolmodel.2015.12.007>. URL [http:](http://)

- 493 [//www.sciencedirect.com/science/article/pii/S030438001500575X](http://www.sciencedirect.com/science/article/pii/S030438001500575X).
- 494 [8] R Core Team. *R: A Language and Environment for Statistical Computing*. R
495 Foundation for Statistical Computing, Vienna, Austria, 2020. URL [https:](https://www.R-project.org/)
496 [//www.R-project.org/](https://www.R-project.org/).
- 497 [9] Martin Plummer, Nicky Best, Kate Cowles, and Karen Vines. *Coda: Output*
498 *Analysis and Diagnostics for MCMC*. 2010.
- 499 [10] Dirk Eddelbuettel and James Joseph Balamuta. Extending R with C++: A
500 Brief Introduction to Rcpp. *PeerJ Preprints*, 5:e3188v1, aug 2017. ISSN 2167-
501 9843. doi: 10.7287/peerj.preprints.3188v1. URL [https://doi.org/10.7287/](https://doi.org/10.7287/peerj.preprints.3188v1)
502 [peerj.preprints.3188v1](https://doi.org/10.7287/peerj.preprints.3188v1).
- 503 [11] Douglas Bates and Dirk Eddelbuettel. Fast and elegant numerical linear alge-
504 bra using the RcppEigen package. *Journal of Statistical Software*, 52(5):1–24,
505 2013. URL <http://www.jstatsoft.org/v52/i05/>.
- 506 [12] Kjell Konis and Florian Schwendinger. *lpSolveAPI: R Interface to lpSolve*,
507 2020. URL <https://CRAN.R-project.org/package=lpSolveAPI>. R package
508 version 5.5.2.0-17.7.
- 509 [13] Jialin Ma, Isuru Fernando, and Xin Chen. *SymEngine: Interface to*
510 *the SymEngine Library*, 2020. URL [https://github.com/symengine/](https://github.com/symengine/symengine.R)
511 [symengine.R](https://github.com/symengine/symengine.R). R package version 0.1.0.
- 512 [14] Robert L. Smith. Efficient Monte Carlo Procedures for Generating Points
513 Uniformly Distributed over Bounded Regions. *Operations Research*, 32(6):
514 1296–1308, December 1984. ISSN 0030-364X. doi: 10.1287/opre.32.6.1296.
- 515 [15] Hans C. Andersen and Persi Diaconis. Hit and Run as a unifying device.
516 *Journal de la Société Française de Statistique*, 148(4):5–28, 2007.

- 517 [16] Aditi Laddha and Santosh Vempala. Convergence of Gibbs Sampling: Coordinate
518 Hit-and-Run Mixes Fast. *arXiv:2009.11338 [cs]*, September 2020.
- 519 [17] M Hakan Satman. RCaller: A software library for calling R from Java. *Journal
520 of Advances in Mathematics and Computer Science*, pages 2188–2196, 2014.
- 521 [18] ICES. Working group on the integrated assessments of the barents sea
522 (WGIBAR). *ICES Scientific Reports*, 2(30):212pp, 2020. URL <https://doi.org/10.17895/ices.pub.5998>.
523
- 524 [19] Ulf Lindstrøm, Benjamin Planque, and Sam Subbey. Multiple patterns of food
525 web dynamics revealed by a minimal non-deterministic model. *Ecosystems*, 20
526 (1):163–182, 2017.
- 527 [20] J.L. Blanchard, J.K. Pinnegar, and S. Mackinson. Exploring marine mammal-
528 fishery interactions using Ecopath with Ecosim: modelling the barents sea
529 ecosystem. Report 117, CEFAS, 2002.

530 8. SIGNIFICANCE STATEMENT

531 9. DATA ARCHIVING STATEMENT

532 10. CONFLICT OF INTEREST STATEMENT

533 11. ETHICS STATEMENT (IF APPLICABLE)

534 12. FUNDING STATEMENT (IF APPLICABLE)

535 13. ACKNOWLEDGEMENTS

536 INRAE, BORDEAUX

537 *Email address:* `hilaire.drouineau@inrae.fr`

538 *Email address:* `Benjamin.Planque@hi.no`

539 *URL:* `www.hi.no`

540 TROMSOE, NORWAY

541 PARIS, FRANCE

542 *Email address:* `christian.mullon@ird.fr`

543 *URL:* `www.ird.fr`