

Theory-driven analysis of ecological data: a practical handbook

us

2024-11-22

Contents

1	About Bookdown	5
1.1	Usage	5
1.2	Render book	5
1.3	Preview book	6
1.4	Here are some useful things for writing the book using bookdown	6
1.5	Cross-references	6
1.6	Parts	7
1.7	Footnotes and citations	8
1.8	Blocks	8
1.9	Sharing your book	9
2	Preamble	11
3	Why mathematical models? (Sonia)	13
3.1	Big data	13
3.2	What is theory?	14
3.3	What's a model?	14
3.4	Purpose of science?	15
3.5	Can we do that with big data analysis?	16
3.6	The scientific method	16
3.7	A feedback loop involving data and models	17
4	What types of theoretical models in ecology? (Isa)	19
4.1	What system? What question? What hypotheses (no maths shown here)	19
4.2	What model formalism? (no maths shown here)	20
4.3	What technical choices?	21
4.4	Some classical models used in ecology and seen next days	23
5	How to build a model? (Isa)	25
6	How to analyze a model? (Isa)	27

7	Temporal data (Day 2)	29
7.1	Introduction	29
7.2	Conclusion	29
8	Spatial data (Day 3)	31
8.1	Introduction	31
8.2	Theoretical frameworks to attack these issues	31
8.3	DATA-DRIVEN (STATISTICAL) APPROACHES	31
8.4	PROCESS-DRIVEN (MECHANISTIC) APPROACHES	32
8.5	Conclusion	32
9	Interaction networks, food webs, and complexity in ecology (Day 4)	33
9.1	Preamble of the script used during the course	33
9.2	Networks and associated definitions	36
9.3	Representation of networks	38
9.4	Crash course: matrices, eigenvalues, Jacobian matrix	42
9.5	Theories and notable results	44
9.6	A quick primer on network statistics	50
9.7	Literature	66

Chapter 1

About Bookdown

This is a *sample* book written in **Markdown**. You can use anything that Pandoc’s Markdown supports; for example, a math equation $a^2 + b^2 = c^2$.

1.1 Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: **# A good chapter**, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: **## A short section** or **### An even shorter section**.

The **index.Rmd** file is required, and is also your first book chapter. It will be the homepage when you render the book.

1.2 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select “All formats” if you’d like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.org/tinytex/>.

1.3 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual `.Rmd` files. You can start the server in a work session by using the RStudio add-in “Preview book”, or from the R console:

```
bookdown::serve_book()
```

1.4 Here are some useful things for writing the book using bookdown

All chapters start with a first-level heading followed by your chapter title, like the line above. There should be only one first-level heading (`#`) per `.Rmd` file.

1.4.1 A section

All chapter sections start with a second-level (`##`) or higher heading followed by your section title, like the sections above and below here. You can have as many as you want within a chapter.

An unnumbered section

Chapters and sections are numbered by default. To un-number a heading, add a `{.unnumbered}` or the shorter `{-}` at the end of the heading, like in this section.

1.5 Cross-references

Cross-references make it easier for your readers to find and link to elements in your book.

1.5.1 Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
 - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world = # Hello world {#hello-world}`.
 - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.

Table 1.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter 1.5.

- If you prefer text as the link instead of a numbered reference use: any text you want can go here.

1.5.2 Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tab:chunk-label)`, respectively.

See Figure 1.1.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Figure 1.1: Here is a nice figure!

Don't miss Table 1.1.

```
knitr::kable(
  head(pressure, 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

1.6 Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in

that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

1.7 Footnotes and citations

1.7.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one ¹.

1.7.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2023] (check out the last code chunk in `index.Rmd` to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file `book.bib`). Note that the `.bib` files need to be listed in the `index.Rmd` with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: <https://rstudio.github.io/visual-markdown-editing/#/citations>

1.8 Blocks

1.8.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (1.1)$$

You may refer to using `\@ref(eq:binom)`, like see Equation (1.1).

1.8.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 1.1.

¹This is a footnote.

Theorem 1.1. *For a right triangle, if c denotes the length of the hypotenuse and a and b denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here <https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html>.

1.8.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>

1.9 Sharing your book

1.9.1 Publishing

HTML books can be published online, see: <https://bookdown.org/yihui/bookdown/publishing.html>

1.9.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

1.9.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book—all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

<https://pkgs.rstudio.com/bookdown/reference/gitbook.html>

Or use:

```
?bookdown::gitbook
```

Chapter 2

Preamble

Who is the textbook for?

Chapter 3

Why mathematical models? (Sonia)

To read: Deep Symbolic Regression for Physics Guided by Units Constraints: Toward the Automated Discovery of Physical Laws <https://iopscience.iop.org/article/10.3847/1538-4357/ad014>

Une intelligence artificielle retrouve des lois physiques à partir de données scientifiques <https://www.insu.cnrs.fr/fr/cnrsinfo/une-intelligence-artificielle-retrouve-des-lois-physiques-partir-de-donnees-scientifiques#:~:text=La%20m%C3%A9thode%20d'intelligence%20artificielle,ces%20r%C3%A8gles%20d'analyse%20dimensionnelle>

3.1 Big data

According to IBM, every day humanity generates 2.5 trillion (2.5 billion billion) bytes of text, image and sound data (<https://www-01.ibm.com/software/fr/data/bigdata>). Acceleration of information accumulation. We can learn from massive amount of data available.

When read by machines (analyses, they can reveal a wealth of unsuspected correlations (or that are difficult to identify). Success of Amazon or Netflix at learning what you like from studying your habits.

Machine learning (ML), a subset of AI that enables computers to learn from training data, has been highly effective at predicting various types of cancer, including breast, brain, lung, liver, and prostate cancer. In fact, AI and ML have demonstrated greater accuracy in predicting cancer than clinicians.

So, knowledge can emerge from this data analysis (even without any idea of the underlying laws).

This lead some people to suggest that we may not need theory anymore (“The end of theory: the data deluge, Wired). The availability of data and the development of methods to analyse them: does they mean that we’re going through a major epistemological change? In other terms: do we still need models and theory?

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2711825/>

3.2 What is theory?

We define *ecological theory* broadly as an explanation of an ecological phenomenon. These explanations take the form of narratives that explain how an ecological process works, or why an ecological pattern is observed, and that becomes scientifically useful when expressed in a logical structure (Pickett et al. 2010; Rossberg et al. 2019).

Maris et al. 2017: “Scientific theories contain universal or general propositions regarding the system in question; they generally encompass a set of models, or rules to build models, as representations of the properties and dynamics of target systems. Models can then be expressed as hypotheses within a formal framework (van Fraassen 1980, Giere 2004).”

The transformation of an idea in narrative form into a logical, testable theory often, though not always, involves the use of *models* (Otto and Rosales 2020).

3.3 What’s a model?

https://en.wikipedia.org/wiki/Ecosystem_model

The concept of a model is derived from scale models, referring to simplified replicas of larger structures such as, for instance, buildings or ships. This implies a structural similarity between a model and its original. Thus, **models are idealized representations of certain aspects of our study systems. They are idealized versions of reality**, just as architectural models represent key features of complex structures and model organisms represent a group of organisms that share common attributes (Kokko 2007).

Models can be as simple as a verbal statement about a subject or two boxes connected by an arrow to represent some relationship (conceptual models). Alternatively, models can be extremely complex and detailed, such as a mathematical description of the pathways of nitrogen transformations within ecosystems.

A mathematical model can be an equation or a set of equations (mathematical expressions) that describes how different aspects of a system relate to one another (Otto and Day 2007). They can be phenomenological or mechanistic (Grainger et al. 2022).

Theory and math are not inexorably linked. Indeed, many excellent theories do not involve math (e.g., the theory of evolution by natural selection; Darwin 1859),

and many uses of math in ecology are not theory (e.g., practical applications of statistics).

Because models are an idealized, simplified version of the real world (e.g. maps), they are therefore not real. They're incomplete, they're wrong (intro Kokko's book). **"all models are false but some models are useful"** https://en.wikipedia.org/wiki/All_models_are_wrong

The systems we're interested in are complex. How can we hope capturing the complexity of reality? We can't. And if we did, it would not be helpful (if models are as complex as reality, e.g. map) because of:

- too many parameters to measure (would require too much time and resources),
- the equations are insoluble and require too long to run on even a good computer,
- even if soluble, we might not be able to understand the results.

Modeling involves a choice about what to include and what to leave out. We need to simplify systems in a way that preserves the essential features of the system (depending on our interest). This means that our big task as modelers is to decide what goes in the model and what doesn't. The art of modeling is to decide which aspect of reality one can sacrifice and which ones are crucial to retain.

We want to work with manageable models which max generality, realism and precision towards the goals of understanding and predicting. . . .

These goals typically compete with each other so real models are mathematical descriptions that result from tradeoffs among these goals which depend on our needs: the tension between realism, generality, and precision: **Levins 1966** <https://v4.chriskycho.com/2016/realism-generality-and-precision-in-tension.html>

- One can sacrifice generality to realism and precision. E.g. fisheries; good measurement of shot-time behavior, numerical solution, precise testable predictions applicable to a particular situation).
- One can sacrifice realism to generality and precision. E.g. physics-like models (general equations such as LV). The way in which nature deviates from theory will indicate where further complexity will be useful.
- One can sacrifice precision to realism and generality. E.g. MacArthur 1965. People concerned with the qualitative behavior (not quantitative). Graphical models.

3.4 Purpose of science?

Understand (and predict).

3.5 Can we do that with big data analysis?

Big data analysis reveals correlations. (Again, this can be useful) But those correlations are not causal. Famous examples of spurious correlations. Correlations can emerge by chance because you look at some many variables. Correlations can happen because of confounding factors, e.g. the example of shoe size and level of math in a school (due to age)

So we can find correlations, but we often don't know why. So we can't understand (understanding is finding associations that are causal).

This does not mean that non-causal correlations can't help predict! They can have great predictive power. Amazon, Netflix. So we can learn from those data and their analysis, even without causal knowledge.

We can predict, but we can also fail at predicting. Predictions can fail because lacks of general rules, so extrapolations can fail; financial crisis, elections. It relies on what happened in the past, so unable to work in new situations.

Predict without understanding... act without understanding? Risks (cf justice, police, discriminations)

Big data don't create theory, they need it to be exploited. (This last statement could evolve in the future? In particular, new analyses methods suggest they can successfully retrieve causality from correlations; cf Zach, Stan's paper with Correlation Cross mapping).

Because of that, we can argue that we still do need theory and models! (models are one language of theory, one way of expressing it)

3.6 The scientific method

Let's step back and reflect on the process by which science is carried out. The *scientific method* is an empirical method for acquiring knowledge that has characterized the development of science since at least the 17th century.

Science (through the scientific method) can build on previous knowledge and develop a more sophisticated understanding of its topics of study over time.

- **'Problem' identification:** It involves careful observation which leads to the formulation of a question.
- **Hypothesis:** A hypothesis is a conjecture (hypothetical explanations), based on the observation/the knowledge obtained while formulating the question, that may explain any given behavior. A scientific hypothesis must be falsifiable, implying that it is possible to identify a possible outcome of an experiment or observation that conflicts with predictions deduced from the hypothesis; otherwise, the hypothesis cannot be meaningfully tested. Falsifiability is a deductive standard of evaluation of scientific theories and hypotheses, introduced by the philosopher of science Karl Popper in his

book *The Logic of Scientific Discovery* (1934). A theory or hypothesis is falsifiable (or refutable) if it can be logically contradicted by an empirical test.

- **Prediction:** The prediction step deduces the logical consequences of the hypothesis before the outcome is known.
- **Testing:** Hypotheses are tested by conducting experiments or gathering observations. The purpose of the test is to determine whether observations agree with or conflict with the expectations deduced from a hypothesis.
- **Refinement (or elimination) of the hypotheses** based on the experimental findings.

Although procedures vary from one field of inquiry to another, the underlying process is frequently the same. In sum, the process is as follows: making conjectures (hypotheses), deriving predictions from them as logical consequences, and then carrying out experiments based on those predictions to determine whether the original conjecture was correct.

The scientific method is an iterative, cyclical process through which information is continually revised.

3.7 A feedback loop involving data and models

A scientific understanding of the biological world arises when ideas about how nature works are formalized, tested, refined, and then tested again.

Scientific inquiry should operate as a feedback loop in which theory that describes the natural world is developed, tested empirically through carefully articulated hypotheses, modified to better represent reality, and then tested again.

When this feedback loop works, **theory** provides a framework to guide inquiry, experimental design, and the interpretation of observed patterns, supplies mathematical tools to harness information from collected data, and connects individual experiments to general ideas about how nature operates.

In turn, **empirical research** can be used to support, refute, or revise theoretical predictions, indicate which theoretical assumptions are consistent with the natural world, and point theoreticians to overlooked processes that can be integrated into models.

However, there is currently a disconnection between theoretical and empirical research.

Although the benefits of feedback between theoretical and empirical research are widely acknowledged by ecologists, this link is still not as strong as it could be in ecological research.

Indeed, up to 45% of articles on empirical ecology make no mention of any theory whatsoever (Scheiner 2013), and fewer than 10% of ecologists and evolutionary biologists agree with the statement that ‘theoretical findings drive empirical work’ in their fields (Haller 2014).

Further references about the disconnect between empirical and theoretical work in ecology: Lomnicki 1988; Kareiva 1989; Fawcett and Higginson 2012; Scheiner 2013; Haller 2014; Rossberg et al. 2019.

Why this disconnection?

- A lack of theoretical training in ecology (Rossberg et al. 2019).
- A lack of motivation on the part of some theoreticians to engage with the language of empiricists (Grimm 1994) or with the elements of nature that empiricists focus on (Krebs 1988).
- A general lack of mutual appreciation between empiricists and theoreticians (Haller 2014).
- Persistent communication barriers between these two groups (Servedio 2020), in particular that theory is expressed in the language of math (some ecologists may not have formal background in math and theoretical papers may not be written with a general audience, not always explaining assumptions, terminology and notations), some aspects of the theory may seem inaccessible by some, the more equations an ecology and evolutionary biology article contains, the fewer citations it receives (Fawcett and Higginson 2012).

This barrier presents a major challenge to the full integration of theoretical and empirical work in ecology.

A better integration of theory into empirical work is needed (Caswell 1988; Pickett et al. 2010; Marquet et al. 2014; Servedio et al. 2014; Servedio 2020).

This is especially important in the context of global change.

This is why this course/book!

Chapter 4

What types of theoretical models in ecology? (Isa)

4.1 What system? What question? What hypotheses (no maths shown here)

- Question is related to a **specific level of organization** / scale and determine what can be ignored
- Examples of questions and processes of interest for each scale based on our recurrent example plant-herbivore (P-H)
- **Updated Levin's triangle** (Bullock 2014) illustrated by different (P-H) models (conceptual, system-specific, generic) with related questions => define variables
- **What can be ignored?** The differences in time scale between processes allow to do some approximations: Upper level is slower than lower ones and its dynamics may be ignored depending on the question. Similarly, lower levels can be considered at equilibrium if much faster than the focal dynamics, or averaged (macroscale averaged view of microscale processes => physics statistics).
- **Link between assumptions, model and predictions**, with the risk of overinterpretation: always keep in mind that what mathematical methods is truth only in the context of initial assumptions.
- **The different types of assumptions** (Servedio 2014):
 - Critical assumptions: the ones necessary to answer the question (structure of the model).
 - Exploratory assumptions: the ones to go deeper in the exploration, assess the robustness of the results.
 - Logistical assumptions: the ones that have to deal with tractability.
- **Illustration of assumption types** with the Grazing-optimisation exam-

ple (De mazancourt et al. 1998):

- Critical assumptions: two paths of recycling with different rates for plants and herbivores.
- Exploratory assumptions: change in the functional response.
- Logistical assumptions: use of deterministic ODEs and continuous time with no space (we'll see all that in more depth in next section).

4.2 What model formalism? (no maths shown here)

4.2.1 Deterministic – stochastic processes

- **What does stochasticity come from?**
 - Environmental (eg: perturbations): does not scale with population size and can vary in space and time.
 - Demographic (eg: more or less offsprings, trait variability that will impact biological rates): scales with population size.
- **When should we account for it** (illustration with pop logistic growth (no math) with addition or not of gaussian noise)
 - for questions related to output variability
 - when output distribution is skewed and mean no more a good predictor (small populations) => when stochastic processes are dominant (ecological or genetical drifts, Allee effect, large demographic stochasticity)
- **When can we use deterministic:** when processes can be summarised with averaged parameters (large populations)
- NB: mention of master equations (deterministic equation summarizing stochastic model)

4.2.2 Time: discrete – continuous

How do the processes are structured along time? when using discrete vs continuous time

- **What are the underlying assumptions?**
 - *Discrete-time:*
 - * Fixed generation times
 - * Synchronization of processes (e.g., seasonal forcing)
 - * Sequential processes with specific order (e.g., complex life-cycle)
 - => more transparent
 - *Continuous time:*
 - * Events can happen at any time
 - * Generation overlap
- **Illustration with logistic pop growth** in continuous and discrete time:
 - discrete-time with small t boils down to continuous

- discrete time have richer dynamics due to a lag
- **Appropriate systems / questions**
 - discrete-time:
 - * questions related to phenology
 - * seasonal dynamics and synchronized generations
 - * complex life cycled
 - continuous-time:
 - * generation overlap
 - * when processes can happen at any time
- Both can be either stochastic or deterministic

4.2.3 Accounting for space

- **None vs spatial:**
 - TIB, example of space implicit; In none spatial models, space is assumed homogenous and the scale can be integrated within the units (example: pop density in ind/km²)
 - When does the question need it explicit space?
 - * When spatial processes impact local dynamics of interest
 - * if we are interested in patterns emerging at higher spatial scale from local processes
- **Topology versus Distances explicit:** does geographical positions matter or only topology?
 - Topology (metapop model): space is represented in the structure of space and strength of connectivity. Adapted to study fragmented landscapes or problems related to structure of connectivity
 - Explicit: distances are explicit; in 1 (Fisher KPP) or two dimensions (grid): adapted to represents fronts, gradients, edge effects or emerging spatial patterns.
- **Discrete within the distance explicit case:** grids vs Continuous (PDE):
 - *Continuous* adapted to model processes occurring progressing locally (diffusion) but not processes with large distance / heterogeneous effects
 - *Grids* with small mesh size can boil down to continuous (sometimes, discretisation is used for a technical approximation of continuous space) => question of technical relevance.
 - As continuous can be more difficult (math) this can prevent its use (mention of lab inheritance)

4.3 What technical choices?

4.3.1 Analytical – Numerical

What do we gain in making models simple? => Principle of parcimony

- **Analytical solution:** we know the state of the model at any time point given $\frac{dN}{dt} \rightarrow f(t)$
- **Tractable model:** We are able to express equilibrium with parameters and study local stability $\frac{dN}{dt} \rightarrow N^* = f(parameters) \Rightarrow$ we can say generalities about the long term dynamics
- **Intractable models:**
 - we are obliged to do simulations using numerical algorithm or integration to approximate the dynamics step by step.
 - the results depend on the parameter values and initial conditions.
 - The dimension of the parameter space to explore is exponentially linked to the number of parameters
 - Simulations allow to investigate transient dynamics

4.3.2 Rules versus Maths

- **What is an agent-based model?** Algorithm which represent processes with a series of rules applied to each agent at each time step using proba: example with birth, death, interactions
- **What is the alternative:** Math \Rightarrow equations: example of ODE, SDE, DE, PDE: what does it mean?
 - ODE: change over time of our variable of interest (SDE; with some stochasticity) \Rightarrow give
 - DE: State at the next time step
 - PDE: Change over time and space
 - Overall idea: Balance between inputs and outputs. See with exponential model?
- **Advantages / Disadvantages:**
 - *ABM:*
 - * The dynamics emerge from elemental processes
 - * Simpler to build from empirical knowledge
 - * But high computation consumption
 - * Rarely tractable
 - *Equations*
 - * Use of math and approximation for simplifications
 - * Have large analysis power for extreme cases
 - * Fast computation
 - * Easier to fit to data
 - * The relations between variables are imposed: processes synthesized
- **To what questions / system/ conditions is it most adapted?**
 - *ABM:*
 - * When stochastic processes are dominant (ex: conservation of small pops)
 - * When there are not too many parameters
 - * For some questions where processes are difficult to synthesise (behaviour)

4.4. SOME CLASSICAL MODELS USED IN ECOLOGY AND SEEN NEXT DAYS23

- * When there is no math skills around!
- *Equations:*
 - * Whenever processes can be synthesised with average parameters
 - * => Large populations
- Such choices are often an inheritance from lab practices and habits.

4.4 Some classical models used in ecology and seen next days

- Systems of differential equations, no space :
 - Verhulst (logistic growth), Lotka-Volterra predator-prey (Day 2)
 - Lotka-Volterra, food web niche model (Day 4)
- Spatial systems of differential equations: TIB, Levins' occupancy (Day3)
- Spatial IBM: Neutral model of biodiversity from Hubbel (Day 3)

A Rmarkdown file is available to provide the code of the illustrations of section 2.2

Chapter 5

How to build a model? (Isa)

(1h) Lecture with interactivity (and Rmarkdown file for some parts)

- I will present different pictures of different plant-herbivore systems with associated research questions and ask how they will sketch the system (on board) to do a model.
- I will ask then which formalism they would choose and why (in terms of determinism, time, space) [and we can try writing rules or equations?]
- Then we will focus on a question for which the Rosenzweig-MacArthur model (ODE) is appropriate, I will write the model on board and ask all the assumptions they could see that are behind the model formulation. We will discuss in which regards they are appropriate for our question.
- Then we will study line by line a code in R to explain how this could be implemented in R, while also explaining the principle of numerical integration.
- I will let them 5-10min to run the model and plot some dynamics and ask them how they would use it to answer the question (to introduce the next section).

Chapter 6

How to analyze a model? (Isa)

(1h) Interactive lecture + based on a script for some parts

(Rmarkdown file with the example of Rosenzweig-MacArthur model)

- Analyze the behavior of the model long-term (Box 3 Grainger et al AmNat)
 - Local stability (when possible for long term dynamics) (script)
 - * Calculating equilibria
 - * What are the Jacobian matrix and eigenvalues (meaning but no math derivation) ?
 - * How to interpret the eigenvalues to infer the long term behavior type
- Isoclines / graphical *Phase plane (script)
- Bifurcation diagrams (long term dynamics) (script)
- When results depend on initial conditions => finding all the equilibria of intractable ODE systems (loop to run large series of initial conditions, function searchZeros of package nleqslv)
- Model Usages to answer the question (will try to show some concrete examples of questions to answer with these usages for the model of section 3a)
 - Parameter variation (see bifurcation diagrams + example DeMaz Grazing-Optimization)
 - Comparison with a null model to assess the impact of a process (with or without the process, with different formulation of the process. Example with functional responses (script))

- Generation of synthetic data on which to run in silico experiments. - Example of food webs under different regimes of perturbations.
- Parameter exploration and robustness of conclusions
 - Exhaustivity is possible (tractable)
 - We know the values of parameters from empirical data => allows us to fix or restrain the range of some parameters.
 - Sensitivity analysis: effect size when varying 10% each parameter + look if conclusions are modified if the most sensitive parameters are varied.

Chapter 7

Temporal data (Day 2)

7.1 Introduction

7.2 Conclusion

Chapter 8

Spatial data (Day 3)

8.1 Introduction

Comapre to Day 2, we will increase scale to consider

Definition of a community and examples

The typical spatial dataset: a community matrix (sites*species), with either presence/absence, or detection/non detection, or abundance, or presence only, data We might also have additional data: species traits, environmental variables, phylogenetic relationships

The typical questions we want to address

8.2 Theoretical frameworks to attack these issues

We present three general frameworks used in this context: – The habitat filtering approach – The metacommunity framework – Vellend’s community ecology theory framework We list strengths/advantages for each framework We point to this cool website for reflections from Vellend and Leibold on the fundamental publications introducing frameworks (2) and (3)

8.3 DATA-DRIVEN (STATISTICAL) APPROACHES

8.3.1 Community level methods

Unconstrained do not directly use environmental variables: a posteriori associations (PCA...) Constrained methods simultaneously consider species composition and environmental descriptors (RDA...) Short focus on Cottenie’s approach

of partitioning variance (partialRDA + PCNM) with examples from mollusk communities in the French Antilles ### Species level methods Permutation approaches, the Cscore, etc.. ==> make a connection with Day 4 here (networks and permutations) We mention the fact that even without interactions, several processes may yield spurious species associations (e.g. Calcagno et al. 2022) Species Distribution Models Occupancy models: their hallmark is to add a detection layer (probability to detect a species when it is present). We will talk more about this in the next section (mechanistic approaches)

8.4 PROCESS-DRIVEN (MECHANISTIC) APPROACHES

8.4.1 Patch-occupancy models

We start by introducing the two simplest and historically important models! Levins (1969) and McArthur & Wilson (1967) models. We highlight the mathematical connection between Levins' model and the logistic model seen on the previous Day There is an intrinsic covariation of r and K in this formulation! We compute the equilibrium occupancies in the two models We introduce the MW model of island biogeography: why are the curves non linear? We make them reflect on this, and on the role of species trait variation in (c,e) We present the example of fitting this model to island presence/absence data: data from Manne et al. (1998) J. Anim. Ecol.

We then discuss possible extensions: – explicit space (IFM models from Hanski and followers) – competition/colonization trade-offs, spatial networks, trophic webs...

Models with explicit abundance dynamics in patches These models quickly become complicated and need to be simulated. One exception is Hubbel's neutral model, which has a dynamics for species abundances in patches and remains reasonably tractable One example of simulation framework for metacommunities is Thompson et al. (2020) Ecology Letters. This is the one we will use to simulate data in the afternoon practical.

8.5 Conclusion

Chapter 9

Interaction networks, food webs, and complexity in ecology (Day 4)

9.1 Preamble of the script used during the course

```
library(cheddar)
library(igraph)

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##   decompose, spectrum

## The following object is masked from 'package:base':
##
##   union

library(vegan)

## Loading required package: permute

##
## Attaching package: 'permute'

## The following object is masked from 'package:igraph':
##
##   permute
```

```
## Loading required package: lattice
## This is vegan 2.6-4
##
## Attaching package: 'vegan'
## The following object is masked from 'package:igraph':
##
##      diversity
library(sbm)
library(alluvial)
library(faux)

##
## *****
## Welcome to faux. For support and examples visit:
## https://debruine.github.io/faux/
## - Get and set global package options with: faux_options()
## *****
library(devtools)

## Loading required package: usethis
##
## Attaching package: 'devtools'
## The following object is masked from 'package:permute':
##
##      check
#Sys.unsetenv("GITHUB_PAT")
#install_github("FMestrel/fw_package")
library(FWebs)
library(matlib)
library(calculus)
library(pracma)

##
## Attaching package: 'pracma'
## The following objects are masked from 'package:calculus':
##
##      cross, gradient, hessian, integral, jacobian, laplacian, taylor
## The following objects are masked from 'package:matlib':
##
##      angle, inv
## The following object is masked from 'package:vegan':
```

```
##
##      procrustes
source('functions_network.R')

####In FWebs, there is a large list of food webs called mg1
sapply(1:length(mg1[[1]]),function(x) dim(as.matrix(mg1[[1]][[x]]))[1])

##      [1] 66 56 50 20 10 16 14 19 48 22 26 24 12 56 46 38 33 26
##      [19] 21 26 22 49 23 28 61 54 30 78 65 29 45 16 41 49 34 12
##      [37] 37 31 78 33 41 78 53 15 12 78 78 24 79 714 35 24 20 30
##      [55] 13 37 72 78 15 28 13 56 27 46 44 16 14 53 40 23 26 68
##      [73] 50 61 21 23 27 25 28 32 19 39 27 57 30 75 15 41 35 39
##      [91] 47 21 50 50 106 16 35 24 29 24 25 21 16 35 6 27 27 28
##     [109] 27 26 52 24 25 26 27 28 13 24 24 44 69 81 39 70 55 18
##     [127] 16 31 17 19 39 32 44 35 65 52 42 32 33 40 14 21 14 35
##     [145] 25 33 26 25 14 39 51 29 24 15 14 27 21 31 41 18 59 49
##     [163] 45 39 35 37 36 27 9 29 32 13 15 7 8 9 36 19 32 22
##     [181] 23 6 20 29 42 57 72 36 31 9 47 89 12 9 12 12 19 24
##     [199] 22 23 8 29 22 22 40 30 7 15 26 90 18 8 27 43 11 32
##     [217] 37 42 45 38 42 37 106 16 35 24 16 36 37 13 9 12 9 12
##     [235] 12 19 24 69 67 55 70 59 24 714 20 22 50 56 29 78 25 22
##     [253] 26 61 48 35 65 78 78 39 10 13 32 31 12 69 57 78 78 16
##     [271] 15 15 72 20 10 22 31 24 25 31 32 20 56 66 17 12 47 30
##     [289] 46 18 8 16 29 50 42 32 31 30 28 56 50 14 25 49 51 75
##     [307] 23 28 36 32 19 34 22 39 68 57 13 36 90 9 7 7 9 47
##     [325] 89 26 34 23 17 14 43 50 61 53 72 35 31 8 13 27 81 53
##     [343] 13 26 30 12 33 41 22 27 28 65 78 37 25 39 29 26 14 25
##     [361] 24 21 15 29 19 27 22 49 30 41 35 44 31 37 22 24 44 42
##     [379] 7 15 26 27 27 24 15 9 39 39 54 49

####Extract data from web 1,223
mat<-as.matrix(mg1[[1]][[223]])
plotMyMatrix(mat)
```

_main_files/figure-latex/unnamed-chunk-5-1.pdf

```
net.mat<-graph_from_adjacency_matrix(mat,mode="directed")
```

TO DO: change the way FWebs is used because it does not compile on Mac and Linux...

9.2 Networks and associated definitions

9.2.1 What is a network?

Networks (or graphs) are sets of nodes linked by edges. Edges represent pairwise relationships between nodes. Edges can be directed or not, weighted or binary (0/1).

9.2.2 Networks in ecology

The main types of networks that you can encounter in ecology: - the most common: interaction networks (edges are interactions, nodes are species) - less common: spatial networks (edges are spatial connections, nodes are populations on a map) - the rarest: assembly networks (edges are transitions between community compositions, nodes are community compositions)

9.2.3 Ecological interaction networks

Since biotic interactions are very diverse, many different types of interaction networks can be built (mutualistic, antagonistic, etc.). The type of data that can be obtained on networks is manifold: - interaction frequency, especially from field observations (e.g. how many times a pollinator species is observed on a particular plant species); - co-abundances or co-occurrences of the species pairs; - interaction potentials, mostly obtained from experiments (e.g. arena experiments in which species are put together pairwise to observe interactions)

9.2.4 A little history

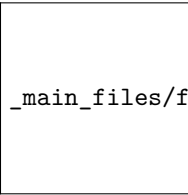
The study of ecological interaction networks begun as early as the 50's with some landmark papers like Hutchinson's (Santa Rosalia, etc.) or Paine's series of keystone predation papers. The book of Allee is remarkable as it contains some of the earliest diagrams of food webs.

9.2.5 Types of network

9.2.5.1 Unipartite networks

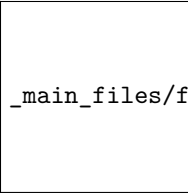
One large class of ecological networks is unipartite networks, i.e. messy ones in which interactions can occur between any two species. E.g. the Benguela food web of Yodzis (1998). The function `sample_gnp` can generate random unipartite networks using the Erdos-Reyi model, which assumes that any interaction has the same probability to exist. The `plot` function of `igraph` makes basic plots of networks, the `layout` option helps see things a wee bit more clearly.

```
net<-sample_gnp(30,0.2,directed = FALSE)
plot(net)
```



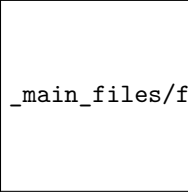
_main_files/figure-latex/unnamed-chunk-6-1.pdf

```
net<-sample_gnp(30,0.2, directed =TRUE)
plot(net)
```



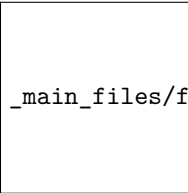
_main_files/figure-latex/unnamed-chunk-6-2.pdf

```
plot(net.mat,layout=layout_with_mds)
```



_main_files/figure-latex/unnamed-chunk-6-3.pdf

```
plot(net.mat,layout=layout_as_tree)
```



_main_files/figure-latex/unnamed-chunk-6-4.pdf

9.2.5.2 Multipartite networks

Another large class of networks is the one consisting of multipartite networks, i.e. networks in which nodes can be partitioned among different levels and nodes only interact among different levels and never within the same level. Each species/node then has a “role” which defines its “level”. The most common type of multipartite network encountered in ecology is bipartite network (only two levels) such as plant-pollinator networks. The function `sample_bipartite` generates a random bipartite network using the Erdos-Renyi model.

```
net<-sample_bipartite(20,20,"gnp",0.1)
plot(net,layout=layout_as_bipartite)
```



```
## [20,] . . . . . 1 . . 1 1 . . . .
## [21,] . . . . . 1 . . . 1 1 1 . 1 . .
## [22,] . . . . . . . . . 1 . . . . .
## [23,] . . . . . . . . . . 1 . . . . .
## [24,] . . . . . . . . . . 1 1 . . . 1 1 1 1 1
## [25,] . . . . . . . . . . 1 1 . . 1 . 1 1 1 1
## [26,] . . . . . . . . . . 1 . . 1 1 . 1 1 1
## [27,] . . . . . . . . . . . . 1 1 1 . 1 . .
## [28,] . . . . . . . . . . 1 . . 1 1 1 1 . 1
## [29,] . . . . . . . . . . . . 1 1 1 . 1 . .
## [30,] . . . . . . . . . . . . . . . . . .
```

When a network is weighted, the weighted adjacency matrix represents not only the existing/absent interactions, but also how strong these interactions are. Here, we create random weights for the edges of network “net” using a Poisson distribution of mean 2. Adding “[.]” after the name of network also yields the adjacency matrix.

```
## 30 x 30 sparse Matrix of class "dgCMatrix"
```

```
## [23,] . . . . . 2 . . . . .
## [24,] . . . . . 5 3 . . . 4 2 2 0 0 .
## [25,] . . . . . 1 1 . . . 4 . 1 1 4 4 .
## [26,] . . . . . 2 . . . 2 1 . 1 3 2 .
## [27,] . . . . . . . . . 2 1 1 . 0 . .
## [28,] . . . . . 1 . . . 0 4 3 0 . 2 .
## [29,] . . . . . . . . . 0 4 2 . 2 . .
## [30,] . . . . . . . . . . . . . . .
```

9.3.3 Network matrices

Quite naturally, if a network is undirected, its adjacency matrix will be symmetric. When a network is directed, it will not necessarily be so. One tricky item to keep in mind: sometimes food webs can be represented not using an adjacency matrix, but rather using an “energy flux” matrix which represents interactions as an antisymmetric matrix of +1’s and -1’s. Adjacency matrices can be binary or real-valued (if they represent interaction weights). A source of confusion can be the difference between adjacency and incidence matrices.

9.3.4 Matrices for multipartite networks

In bipartite networks, the diagonal blocks of the adjacency matrix are full of 0 only. It is thus more useful to only look at the (rectangle) non-zero off-diagonal blocks, which are called the incidence matrix in ecology (or bi-adjacency matrix in other disciplines). The function `as_biadjacency_matrix` of `igraph` extracts this matrix from a bipartite network.

```
net<-sample_bipartite(4,4,"gnm",m=8)
net[,]

## 8 x 8 sparse Matrix of class "dgCMatrix"
##
## [1,] . . . . 1 1 . .
## [2,] . . . . 1 1 1 .
## [3,] . . . . . 1 1
## [4,] . . . . 1 . . .
## [5,] 1 1 . 1 . . . .
## [6,] 1 1 . . . . .
## [7,] . 1 1 . . . . .
## [8,] . . 1 . . . . .

as_biadjacency_matrix(net)

##      5 6 7 8
## 1 1 1 0 0
## 2 1 1 1 0
## 3 0 0 1 1
```



```
## 4 1 0 0 0
```

9.3.5 Degrees

A basic element of representation of networks is the degree of a node. The degree of a node is the number of interactions it shares with other nodes. For undirected networks, this definition is straightforward and corresponds to row- or column-sums of the binary adjacency matrix. For directed networks, one actually needs to define the in-degree and out-degree of each node, which correspond respectively to the number of incoming and outgoing edges. The function `degree` of `igraph` computes all of these using the argument called `mode`.

```
degree(net)
```

```
## [1] 2 3 2 1 3 2 2 1
```

```
degree(net,mode="in")
```

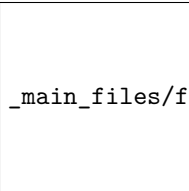
```
## [1] 2 3 2 1 3 2 2 1
```

```
degree(net,mode="out")
```

```
## [1] 2 3 2 1 3 2 2 1
```

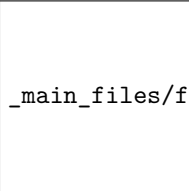
The degree distribution corresponds to the realized empirical distribution of degrees in the network. It can be obtained using a variety of functions, including `degree_distribution` in `igraph`.

```
hist(degree(net.mat),breaks=0:max(degree(net.mat)))
```



```
_main_files/figure-latex/unnamed-chunk-12-1.pdf
```

```
plot(degree_distribution(net.mat, cumulative = TRUE))
```



```
_main_files/figure-latex/unnamed-chunk-12-2.pdf
```

9.3.6 Connectance

Connectance is an important notion. It is defined as the proportion of possible interactions that really exist. There are plenty of ways to implement its compu-

tation (e.g. using mean), but it is important to correct these according to the impossible interactions (e.g. correcting for self-links or edges within levels in bipartite networks).

```
conn<-mean(as.matrix(net.mat[,]))
conn
```

```
## [1] 0.02669989
```

9.4 Crash course: matrices, eigenvalues, Jacobian matrix

9.4.1 Crash course: matrices

Matrices are mathematical objects that represent tables, i.e. have rows and columns. The element at row i and column j of matrix M is often noted m_{ij} . Matrices and vectors follow a strange multiplication operation, noted with `%%` in *R* (rather than `*` which encodes a more plastic multiplication operation).

```
m<-matrix(rbinom(9,1,0.5),nrow=3)
x<-rnorm(3)
m*x
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.09299545 0.09299545 0.00000000
## [2,] 0.24709379 0.00000000 0.2470938
## [3,] 0.01487206 0.00000000 0.00000000
```

```
m%%x
```

```
##           [,1]
## [1,] 0.34008924
## [2,] 0.10786751
## [3,] 0.09299545
```

9.4.2 Crash course: matrix eigenvalues

All matrices are associated with a set of values called “eigenvalues” which represent the “multiplicators” of the matrix. Each of these eigenvalues λ is associated to eigenvectors x such that multiplying x by the matrix M results in only multiplying x by λ :

$$M.x = \lambda x$$

The collection of eigenvalues of M is called its spectrum. The spectrum is always finite and has at most a number of distinct elements equal to the smallest dimension of M .

9.4. CRASH COURSE: MATRICES, EIGENVALUES, JACOBIAN MATRIX 43

The function `eigen` yields the eigensystem (eigenvalues and eigenvectors) of a matrix.

```
eigen(m)
```

```
## eigen() decomposition
## $values
## [1] 1.8392868+0.0000000i -0.4196434+0.6062907i -0.4196434-0.6062907i
##
## $vectors
##           [,1]           [,2]           [,3]
## [1,] -0.7071068+0i  0.4024298+0.1718667i  0.4024298-0.1718667i
## [2,] -0.5934654+0i -0.6755079+0.0000000i -0.6755079+0.0000000i
## [3,] -0.3844462+0i -0.1189573-0.5814209i -0.1189573+0.5814209i
```

9.4.3 Crash course: Jacobian matrices

A set of ordinary differential equations (ODEs), such as those given by Lotka-Volterra equations, can be written in compact form as:

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x})$$

Let us assume that there exists an equilibrium \vec{x}^* for this system. If we want to assess whether this equilibrium is stable, we need to linearize the dynamics around the equilibrium:

$$\frac{d(\vec{x} - \vec{x}^*)}{dt} \approx J(\vec{x}^*) \cdot (\vec{x} - \vec{x}^*)$$

The matrix $J(\vec{x}^*)$ is called the Jacobian matrix of the dynamical system at the equilibrium \vec{x}^* . The expression for the Jacobian matrix is obtained using the partial derivatives of the different components of \vec{F} :

$$J(\vec{x}^*) = (\partial F_i / \partial x_j)|_{x=x^*}$$

Formal computations of Jacobian matrices can be done with R using the package `calculus`:

```
#jacobian(c("r*n1*(1-n1/k)-a*n1*n2/(1+h*n1)", "b*n1*n2/(1+h*n1)-d*n2"), var = c("n1", "n2"))
```

9.5 Theories and notable results

9.5.1 Lotka-Volterra and networks

This is the classic Lotka-Volterra system, given for a dimensionless system (i.e. the coefficient of self-feedback are all set to 1):

$$\frac{dx_i}{dt} = x_i \left[r_i - x_i + \sum_j a_{ij} x_j \right]$$

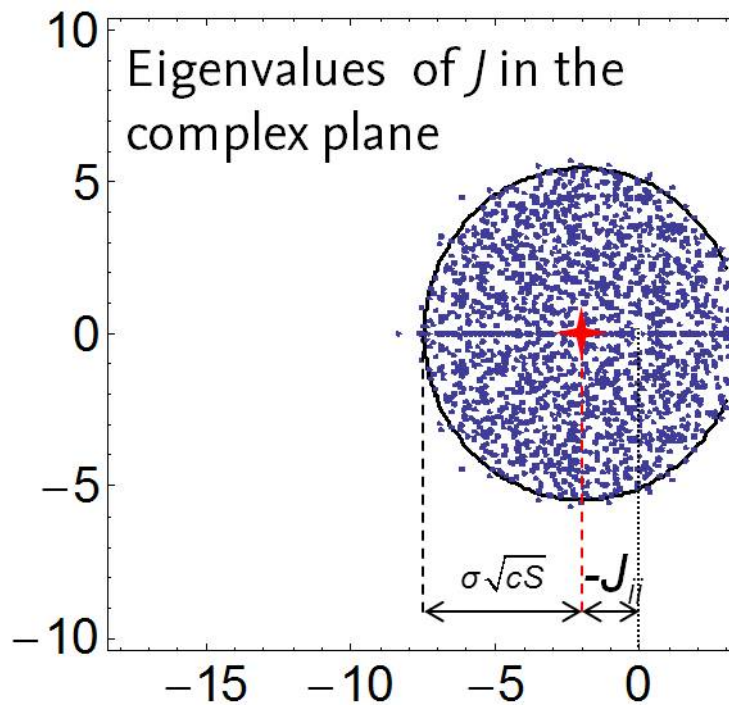
This system of ODEs can be seen as describing the dynamics of a network of species, with matrix A as its “weighted” adjacency matrix.

9.5.2 May’s stability result

In the 70’s, Sir Robert May published a very important mathematical ecology result which shook some of the beliefs ecologists had at that time. The idea is to assume that a community consists of S species and that species abundances of these species are fixed at an equilibrium value \vec{x}^* . May’s approach asks the question: what should happen if elements of the Jacobian matrix were randomly drawn according to a simple distribution? More precisely, May assumed that diagonal elements of the Jacobian were equal (or similar) and negative, while the off-diagonal elements had a probability c (the connectance) of being non-zero, and if not equal to zero, followed a normal distribution of mean 0 and standard deviation σ . The result of May thus stipulates that the equilibrium can only be stable when

$$\sigma\sqrt{cS} < -\overline{J_{ii}} \quad (9.1)$$

This result can be explained geometrically (based on the circular law): for a random matrix with this particular distribution of off-diagonal elements, the distribution of eigenvalues will be enclosed in a disk of center $(-\overline{J_{ii}}, 0)$ and radius $\sigma\sqrt{cS}$ in the complex plane. For all eigenvalues to have negative real parts, the radius of this disk should be less than the distance between $(0,0)$ and its center.



```
m<-matrix(rnorm(10^6),nrow=10^3)
plot(eigen(m)$values,asp=1)
```

_main_files/figure-latex/unnamed-chunk-18-1.pdf

This result can be interpreted as follows: for a given equilibrium to be stable, it should have low diversity and/or low connectance and/or a low variability of non-diagonal Jacobian elements (i.e. feedbacks among species abundances) and/or highly negative diagonal Jacobian elements (i.e. strong negative feedbacks of species upon themselves).

Intuitively, this does not seem to conform to reality: we see species-rich systems in which species interact relatively strongly and not too sparsely, without intense self-regulation... So, some of the assumptions must be false and we can look for changes to these assumptions: - dynamics could be transient rather than at equilibrium - interactions might be structured and not completely random (see Claire's talk)

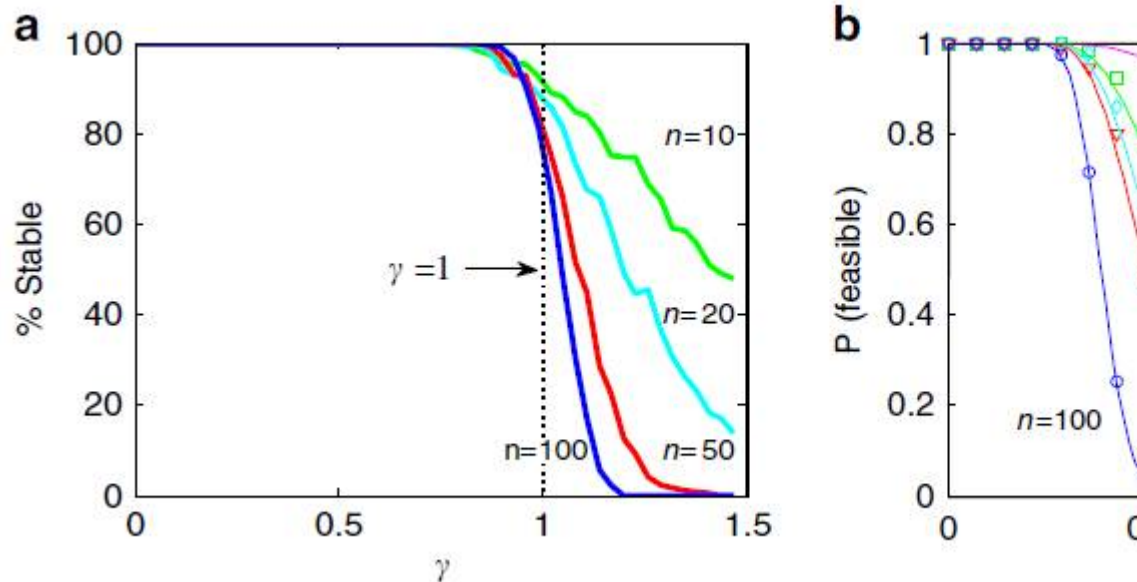
9.5.3 Feasibility vs. stability

One other strong argument contradicting May's result is the following: in many cases, it might actually be more difficult to obtain an equilibrium including all species (i.e. have feasibility) than for this equilibrium to be stable. For instance, Bizeul & Najim proved that in a certain class of LV systems, feasibility is obtained only when

$$\sigma\sqrt{2S\log(S)} < -\overline{J_{ii}}$$

where σ is the standard deviation of the LV matrix (not the Jacobian matrix) and the connectance is merged into σ^2 .

In the simulation paper of L. Stone, similar results were obtained – here stability (on the left) is lost after feasibility (on the right) when increasing of a parameter (γ) which is proportional to our σ .



9.5.4 Food web invariants

In the 80-90's, one important goal of interaction network research was to find invariant properties, i.e. statistics or metrics that did not vary among food webs. With this goal in mind, two models were proposed. First, the cascade model of Cohen & Briand proposed that predators only eat a rather fixed number of prey species. This corresponds to predators eating prey that are smaller than them with probability c/S (c being a fixed parameter), so that connectance decreases as $1/S$.

```
m<-cascade_matrix(10,20)
sum(m)/(dim(m)[1]*(dim(m)[1]-1))
```

```
## [1] 0.2710526
m<-cascade_matrix(10,200)
sum(m)/(dim(m)[1]*(dim(m)[1]-1))
```

```
## [1] 0.02389447
```

The second model that was developed to find invariants is the niche model of Williams and Martinez. In this model, all species are supposed to be defined by their size and to eat all species with size contained within an interval centred around a size lower than the predator's. With the right parameter values, one can use this model to generate a food web that has a given connectance, whatever the number of species in the food web.

```
niche<-niche_matrix(0.2,100)
m<-niche$matrix
sum(m)/(dim(m)[1]^2)
```

```
## [1] 0.1799
niche<-niche_matrix(0.2,200)
m<-niche$matrix
sum(m)/(dim(m)[1]^2)
```

```
## [1] 0.198325
```

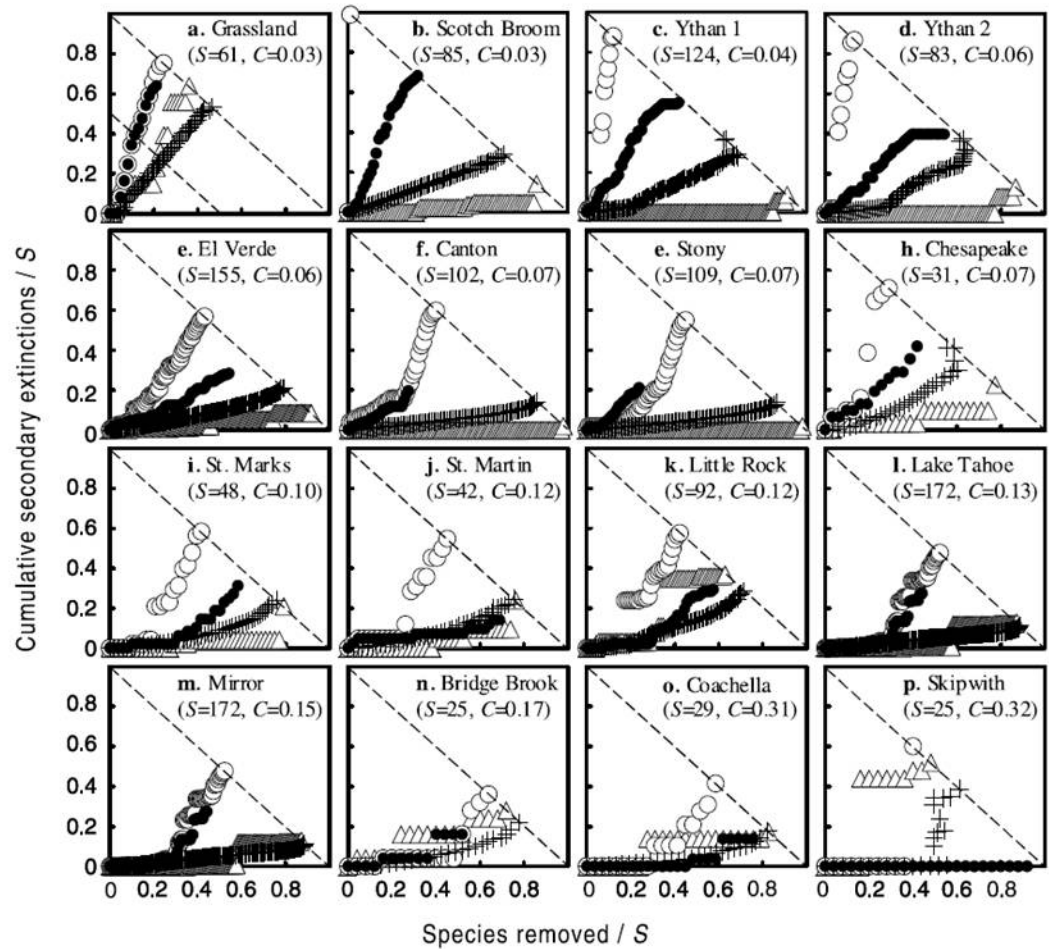
9.5.5 Generating the niche model

Exercise: generate 100 virtual food webs similar to the one described by mat

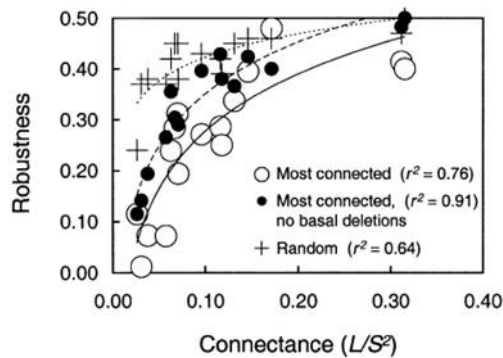
```
niches<-lapply(1:100,function(x) niche_matrix(conn,dim(mat)[1]))
ms<-lapply(1:100,function(x) niches[[x]]$matrix)
```

9.5.6 Robustness to secondary extinctions

The general idea is to assume that when a species is removed from a network, the species interacting with it could go extinct (secondary extinction). Robustness analysis then consists in assessing how many species are lost when R species are removed. The most used indicator of robustness is the R_{50} which is the percentage of species one needs to remove from a web to lose 50% of its species.

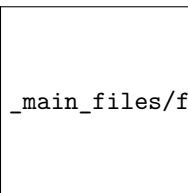


In the paper of J. Dunne and collaborators, they used examples based on a variety of datasets. The basic rule then is that a species goes extinct when it loses all its prey items (except itself if it is cannibalistic). One can obtain different scenarios based on how removed species are chosen (intentionality: how strongly the species' degree changes its probability of being drawn randomly as the next one to be removed).



```
net<-graph_from_adjacency_matrix(m,mode="directed")
i_index <- seq(from = 0, to = 1, by =0.1)
i_index <- head(i_index,-1)
prob_exp<-exponent.removal(net, i_index)
V(net)$name<-1:200
iterate(fw_to_attack=net, prob_exp, alpha1=50, iter=10, i_index, plot = TRUE)
```

```
## Iteration 1
## Iteration 2
## Iteration 3
## Iteration 4
## Iteration 5
## Iteration 6
## Iteration 7
## Iteration 8
## Iteration 9
## Iteration 10
```



_main_files/figure-latex/unnamed-chunk-25-1.pdf

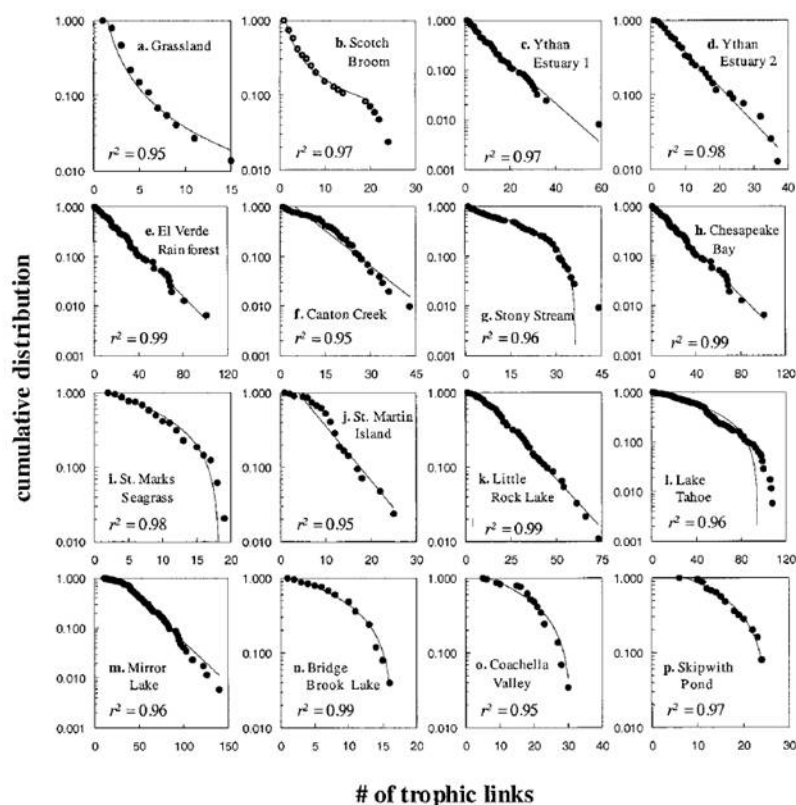
```
##      i_index  meanR      sdR lower.bound upper.bound
## 1      0.0 0.5025 0.002635231 0.5008667 0.5041333
## 2      0.1 0.5050 0.000000000 0.5050000 0.5050000
## 3      0.2 0.5050 0.000000000 0.5050000 0.5050000
```

## 4	0.3	0.5050	0.000000000	0.5050000	0.5050000
## 5	0.4	0.5050	0.000000000	0.5050000	0.5050000
## 6	0.5	0.5050	0.000000000	0.5050000	0.5050000
## 7	0.6	0.5050	0.000000000	0.5050000	0.5050000
## 8	0.7	0.5050	0.000000000	0.5050000	0.5050000
## 9	0.8	0.5050	0.000000000	0.5050000	0.5050000
## 10	0.9	0.5050	0.000000000	0.5050000	0.5050000

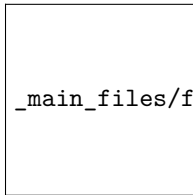
9.6 A quick primer on network statistics

9.6.1 Degree distributions

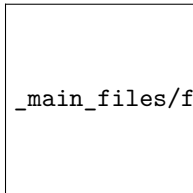
Basically, one can do three things with degree distributions: - compute them empirically - generate networks similar to one that is observed using its degree sequence - compare the degree distribution to benchmark distributions of integers (Poisson, power law, ...) or of degrees of well-known random networks (Erdos-Renyi, random geometric graph, Watts-Strogatz...)



```
net<-graph_from_adjacency_matrix(m)
hist(degree(net),breaks=0:max(degree(net)))
```



```
plot(degree_distribution(net, cumulative = TRUE))
```



```
net<-sample_degseq(degree(net),method = "v1")
mean(degree(net))
```

```
## [1] 79.33
```

```
ks.test(degree(net),"pbinom",length(V(net)),mean(degree(net))/length(V(net)))
```

```
## Warning in ks.test.default(degree(net), "pbinom", length(V(net)),
## mean(degree(net))/length(V(net))): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: degree(net)
## D = 0.32724, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

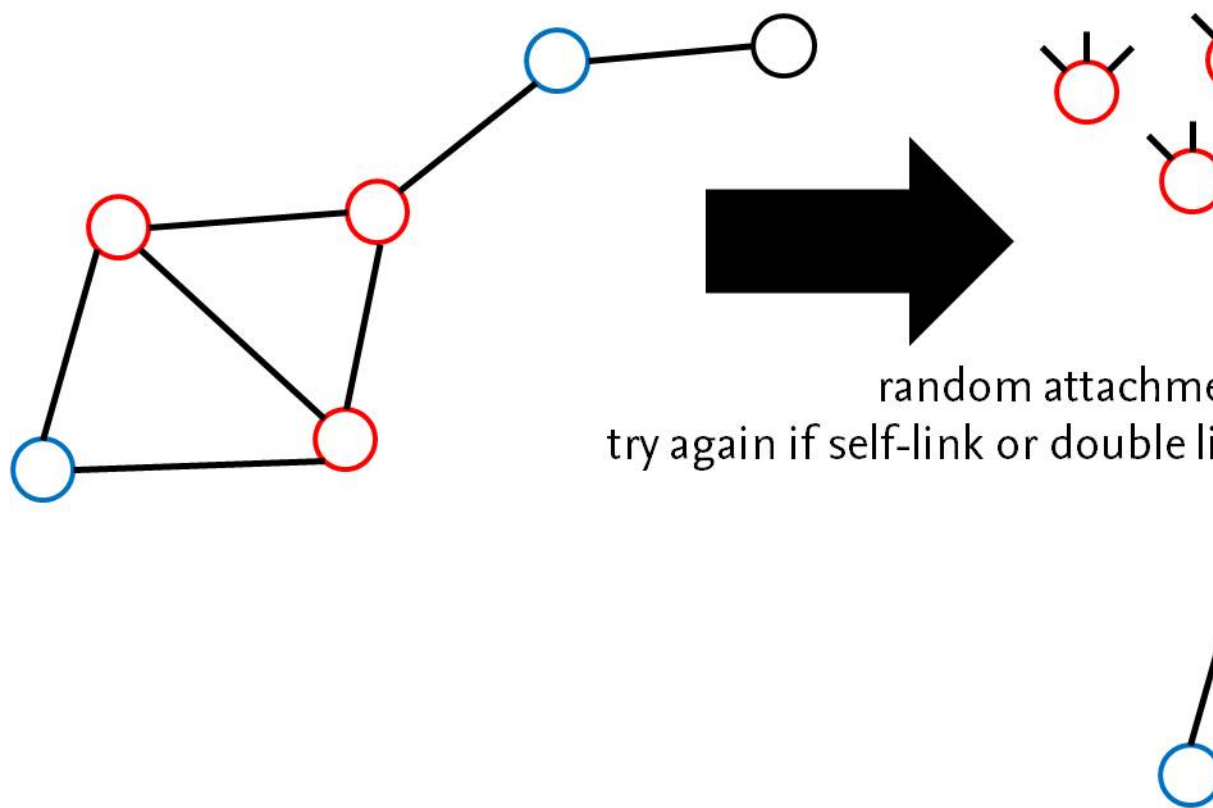
Power laws are among the most popular distributions of degrees discussed in the literature. However, a question worth asking is whether power laws are useful in ecology, i.e. whether it is actually possible to ascertain power-lawed degrees in ecological networks. As Stumpf & Porter (2012) clearly articulated, this would require to have degrees ranging at the very least between 1 and 100 (2 orders of magnitude), which is quite unheard of.

9.6.2 Null models / randomizations

Statistics on networks are hard to test because everything depends on everything else in a network...

So to test for extraordinary patterns in a network (or between them), one can resort to two strategies: - either to define a probabilistic model to build a null hypothesis (e.g. using the expected degree distribution) - or to define a null model based on some randomization scheme

One of the most useful and simplest such scheme is the configuration model. Under this null model, edges are randomly reattached to other nodes such that all nodes keep their degrees. This behaves effectively as if all nodes were attached to half-edges which were randomly paired until the resulting network yields no self-link and no double link.



For bipartite binary networks, there is an algorithm (curveball) which yields the configuration model quickly and efficiently.

```
net<-sample_bipartite(50,50,"gnp",0.1)
sample.bip.config<-simulate(nullmodel(as_incidence_matrix(net),"curveball"),nsim=1000)
```

```
## Warning: `as_incidence_matrix()` was deprecated in igraph 1.6.0.
## i Please use `as_biadjacency_matrix()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
dim(sample.bip.config)
```

```
## [1] 50 50 1000
```

For unipartite binary networks, the task is slightly harder, but doable using `sample_degseq`.

```
n<-200 #1000
net<-sample_gnp(n,0.2, directed = FALSE)
sample.config.undirected<-lapply(1:100,function(x) sample_degseq(degree(net), method = "v1"))
length(sample.config.undirected)
```

```
## [1] 100
```

For directed networks, it is even worse because we have no indication that the random matrices are all explored and represented fairly among the realizations of the algorithm.

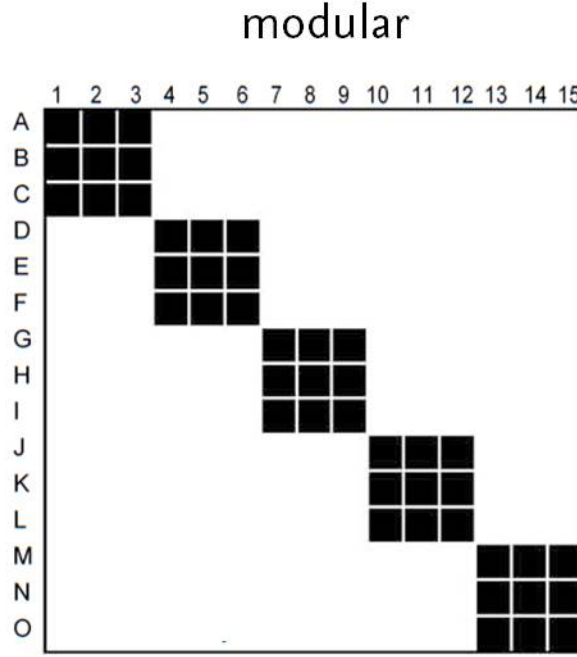
```
net<-sample_gnp(n,0.2, directed = TRUE)
sample.config.directed<-lapply(1:100,function(x) sample_degseq(degree(net,mode="out"), degree(net)
length(sample.config.directed)
```

```
## [1] 100
```

Finally, for any weighted network (bipartite or unipartite), the task is even more difficult.

9.6.3 Network structure

Ecological interaction networks can have non-random structures. In ecology, we tend to focus on two such structures: modularity (on the left) and nestedness (on the right).



In the following we will more strongly focus on modularity than on nestedness, for reasons that will be explained later.

9.6.4 Modularity

Modularity, as defined by Newman, is given by the following formula:

$$Q = \frac{1}{A} \sum_{i,j} \left[a_{ij} - \frac{d_i d_j}{A} \right] \delta_{ij} \quad (9.2)$$

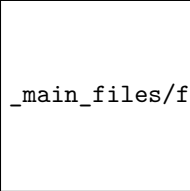
where A is the total number of connections in the network (i.e. $A = \sum_i \sum_j a_{ij}$), d_i is the degree of species i (i.e. $d_i = \sum_j a_{ij}$) and the δ_{ij} are dummy variables indicating whether species/nodes i and j are assumed to belong to the same module/community/cluster. In layman's terms, modularity is a quantity that measures how "intense" interactions are within modules, and thus by contrast how little interaction there is between nodes belonging to different modules. The principle of modularity search is to look for the partitioning of nodes into communities/modules that maximizes the associated modularity score. The problem of modularity search is quite complex (NP-complete problem) and thus can be solved with different algorithms that have their pros and cons.

Modularity, in its classic form, only works for undirected networks. Three algorithms that are often used in ecology are the edge-betweenness algorithm

(EB), the leading-eigenvector one (LE) and the Louvain algorithm (ML).

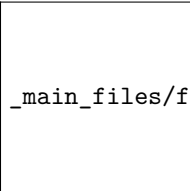
```
net<-sample_gnp(100,0.2, directed = FALSE)
EB.mod<-cluster_edge_betweenness(net)
LE.mod<-cluster_leading_eigen(net)
ML.mod<-cluster_louvain(net)

plot(EB.mod,net,layout = layout_with_mds)
```



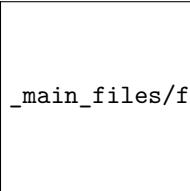
_main_files/figure-latex/unnamed-chunk-33-1.pdf

```
plot(LE.mod,net,layout = layout_with_mds)
```



_main_files/figure-latex/unnamed-chunk-33-2.pdf

```
plot(ML.mod,net,layout = layout_with_mds)
```



_main_files/figure-latex/unnamed-chunk-33-3.pdf

Again, modularity is not adapted to directed networks, but there ways to circumvent this issue: - by making the network/adjacency matrix symmetric - through the use of another definition of modularity (e.g. the infomap definition of Rosvall and Bergström)

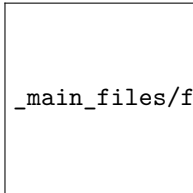
9.6.4.1 Modularity of the empirical food web

As mentioned before, it is highly recommended to check whether the obtained modularity is “expected” or extraordinary using a null model. Here is an example using the empirical food web.

```
modul<-cluster_louvain(graph_from_adjacency_matrix(mat,mode="undirected"))
```

```
## Warning: The `adjmatrix` argument of `graph_from_adjacency_matrix()` must be symmetric
## with mode = "undirected" as of igraph 1.6.0.
```

```
## i Use mode = "max" to achieve the original behavior.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
moduls<-lapply(1:100,function(x) cluster_louvain(graph_from_adjacency_matrix(ms[[x]],m
plot(density(unlist(moduls)))
```



```
modul.ecdf<-ecdf(unlist(moduls))
1-modul.ecdf(modul$modularity[2])
```

```
## [1] 1
```

9.6.4.2 Which algorithm?

Because there are tons of algorithms already developed to find clusters/communities/modules in networks, it is easy to get lost... Yet two papers (at least) have made our life easier by comparing all these algorithms, both for unipartite networks (Yang et al. 2016) and bipartite networks (Leger et al. 2015).

9.6.5 Block models

Another possibility when looking for clusters of species that interact more among themselves is to look for blocks of nodes/species that “behave” similarly. This can be achieved using stochastic block models (SBM), also sometimes called latent block models (LBM) when dealing with bipartite networks.

Block models work by optimizing the fit of a model (in the classic statistical sense) which assigns probability of membership to groups and probability of interaction between nodes of known groups. Block models can be fitted on different kinds of adjacency matrices following different distributions (e.g. Bernoulli for binary adjacency matrices, Poisson for counts, etc.). The inference of a block model yields one parameter per block (the “expected connection” within the block) and the proportions of nodes belonging to each group.

```
sbmnet <- sampleSimpleSBM(100, c(.5, .25, .25), list(mean = diag(.4, 3) + 0.05), model
head(sbmnet$networkData)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]  NA   0   0   0   0   0   1   0   0   0   1   0   1   0
## [2,]   0  NA   0   0   0   0   1   0   0   0   0   0   0   0
```



```

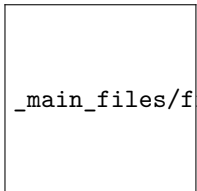
## [3,] 0 0 NA 0 0 0 0 0 0 1 0 0 0 0
## [4,] 0 0 0 NA 1 0 0 1 0 0 0 0 0 0
## [5,] 0 0 0 1 NA 1 1 0 0 0 0 0 0 0
## [6,] 0 0 0 0 1 NA 0 0 1 0 0 0 0 0
##      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
## [1,] 0 0 0 1 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 1 0 0 0
## [3,] 0 0 0 0 0 0 0 1 0 0 0 0
## [4,] 1 0 0 0 0 0 0 0 1 0 1 0
## [5,] 0 0 0 0 0 0 1 0 0 0 0 0
## [6,] 0 0 0 0 0 0 1 1 0 0 0 0
##      [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
## [1,] 1 0 0 0 0 1 0 0 1 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 1 0 0 0 0 0 0
## [4,] 0 0 0 0 0 1 0 0 0 0 0 0
## [5,] 0 1 0 1 0 0 0 0 0 0 1 0
## [6,] 0 1 0 1 0 0 0 0 0 1 1 0
##      [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
## [1,] 0 0 0 0 0 0 1 0 0 0 0 0
## [2,] 0 0 0 0 1 0 0 0 0 0 0 0
## [3,] 1 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 1 0 0 0 0 0 1 1
## [5,] 0 0 1 1 0 0 0 0 0 0 0 0
## [6,] 0 1 0 0 0 0 0 0 0 0 1 0
##      [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61] [,62]
## [1,] 0 0 0 1 0 0 0 0 0 0 0 0
## [2,] 1 0 0 0 0 0 0 1 1 0 0 1
## [3,] 0 0 0 1 1 0 0 0 0 0 0 0
## [4,] 0 0 0 1 0 0 0 0 0 0 0 1
## [5,] 0 0 0 0 0 0 0 0 1 0 1 1
## [6,] 0 0 0 0 0 1 0 0 1 0 0 0
##      [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72] [,73] [,74]
## [1,] 0 0 0 0 0 0 0 0 0 1 0 0
## [2,] 0 0 0 0 0 0 0 1 0 0 1 0
## [3,] 0 0 0 1 0 0 0 0 0 1 0 0
## [4,] 0 0 0 0 1 0 0 0 0 0 1 0
## [5,] 0 0 0 0 1 0 0 0 0 0 0 0
## [6,] 0 0 0 1 1 0 0 0 1 0 0 0
##      [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85] [,86]
## [1,] 0 0 0 1 0 1 0 0 0 1 0 0
## [2,] 1 0 0 0 0 0 0 0 0 0 1 0
## [3,] 0 0 0 1 0 0 0 0 0 1 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 1 0 0 0 0 1
## [6,] 0 0 0 0 0 0 1 0 0 1 0 0

```

```
##      [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97] [,98]
## [1,]    0    0    0    0    0    0    0    0    0    0    0    0
## [2,]    0    0    0    0    0    0    0    0    0    1    1    0
## [3,]    0    0    0    0    1    0    0    0    0    1    1    0
## [4,]    0    0    0    0    0    0    0    0    0    0    0    1
## [5,]    0    0    0    0    0    0    0    0    0    0    0    1
## [6,]    0    0    0    0    0    0    1    0    1    0    0    1
##      [,99] [,100]
## [1,]    0      1
## [2,]    0      0
## [3,]    0      0
## [4,]    0      0
## [5,]    0      0
## [6,]    0      0
```

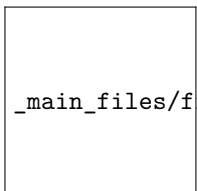
```
net.SBM <- estimateSimpleSBM(as.matrix(sbmnet$networkData))
```

```
## -> Estimation for 1 groups
##
```




_main_files/figure-latex/unnamed-chunk-35-1.pdf

```
## -> Computation of eigen decomposition used for initializations
##
## -> Pass 1
##      -> With ascending number of groups
##      -> For 2 groups
##
```




_main_files/figure-latex/unnamed-chunk-35-2.pdf

```
##      -> For 3 groups
##
```




_main_files/figure-latex/unnamed-chunk-35-3.pdf

-> For 4 groups
##




_main_files/figure-latex/unnamed-chunk-35-4.pdf

-> For 5 groups
##




_main_files/figure-latex/unnamed-chunk-35-5.pdf

-> With descending number of groups
-> For 4 groups
##



_main_files/figure-latex/unnamed-chunk-35-6.pdf

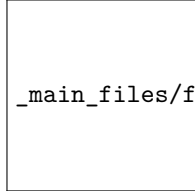
-> For 3 groups
##



_main_files/figure-latex/unnamed-chunk-35-7.pdf

-> For 2 groups

##



_main_files/figure-latex/unnamed-chunk-35-8.pdf

-> Pass 2

-> With ascending number of groups

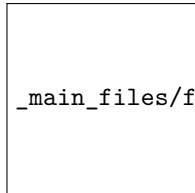
-> For 2 groups

-> For 3 groups

-> For 4 groups

-> For 5 groups

##

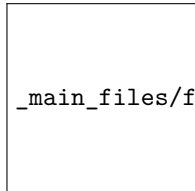


_main_files/figure-latex/unnamed-chunk-35-9.pdf

-> With descending number of groups

-> For 4 groups

##



_main_files/figure-latex/unnamed-chunk-35-10.pdf

-> For 3 groups

-> For 2 groups

-> Pass 3

-> With ascending number of groups

-> For 2 groups

-> For 3 groups

-> For 4 groups

-> For 5 groups

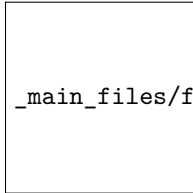
-> With descending number of groups

-> For 4 groups

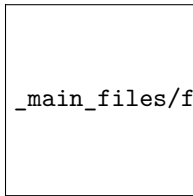
-> For 3 groups

-> For 2 groups

```
plot(net.SBM, 'expected')
```



```
plot(net.SBM, 'data')
```



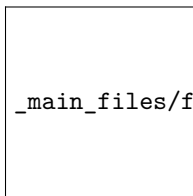
9.6.6 Modularity and blocks

It is possible to compare the partitionings of nodes obtained through modularity search and SBMs, for instance comparing the result of Louvain algorithm with a simple SBM on the network given by `mat`.

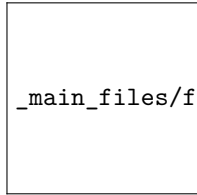
The alluvial plot helps visualize the correspondence between these groups.

```
m.SBM <- estimateSimpleSBM(mat)
```

```
## -> Estimation for 1 groups
##
```

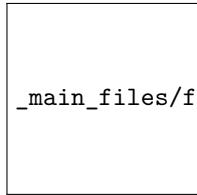


```
## -> Computation of eigen decomposition used for initializations
##
## -> Pass 1
##   -> With ascending number of groups
##   -> For 2 groups
##
```



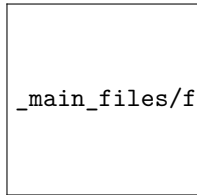
_main_files/figure-latex/unnamed-chunk-36-2.pdf

-> For 3 groups
##



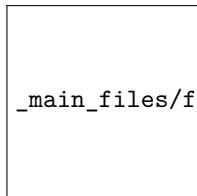
_main_files/figure-latex/unnamed-chunk-36-3.pdf

-> For 4 groups
##



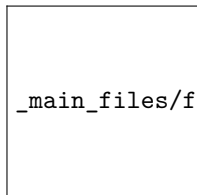
_main_files/figure-latex/unnamed-chunk-36-4.pdf

-> For 5 groups
##



_main_files/figure-latex/unnamed-chunk-36-5.pdf


-> With descending number of groups
-> For 4 groups
##



_main_files/figure-latex/unnamed-chunk-36-6.pdf

-> For 3 groups


```
##
```



```
_main_files/figure-latex/unnamed-chunk-36-7.pdf
```

```
##          -> For 2 groups
```

```
##
```



```
_main_files/figure-latex/unnamed-chunk-36-8.pdf
```

```
## -> Pass 2
```

```
##          -> With ascending number of groups
```

```
##          -> For 2 groups
```

```
##          -> For 3 groups
```

```
##          -> For 4 groups
```

```
##          -> For 5 groups
```


```
##          -> With descending number of groups
```

```
##          -> For 4 groups
```

```
##          -> For 3 groups
```

```
##          -> For 2 groups
```

```
modul<-cluster_louvain(graph_from_adjacency_matrix(mat,mode="undirected"))
make_alluvial_2(m.SBM$memberships,modul$membership,"Blocks","Modules")
```



```
_main_files/figure-latex/unnamed-chunk-36-9.pdf
```

It is also possible to look at whether these groupings have anything to do with trophic level (sensus MacKay et al.).

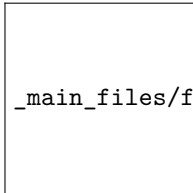
```
count_components(net.mat)
```

```
## [1] 4
```

```
net.comp<-components(net.mat)
```

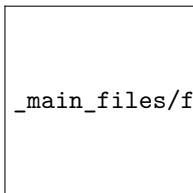
```
tl.1<-trophic_levels(largest_component(net.mat))
```

```
plot(largest_component(net.mat),layout = layout_as_food_web)
```



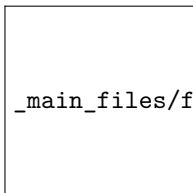
_main_files/figure-latex/unnamed-chunk-37-1.pdf

```
plot(tl.1~as.factor(m.SBM$indMemberships[which(net.comp$membership==1),1]*%(1:3)),xlab="module",ylab="trophic level")
```



_main_files/figure-latex/unnamed-chunk-37-2.pdf

```
plot(tl.1~as.factor(modul$membership[which(net.comp$membership==1)]),xlab="module",ylab="trophic level")
```



_main_files/figure-latex/unnamed-chunk-37-3.pdf

As we can see, SBM-derived groups seem to be better at capturing trophic levels.

9.6.7 Spectral clustering

One drawback of SBM is that they can be slow to converge, especially when the dataset is big. A convenient replacement algorithm when the number of groups to look for is known (or can be easily guessed) is spectral clustering. The idea of spectral clustering is to make use of the Laplacian matrix L of the graph, given by:

$$L = D - A$$

where D is the diagonal matrix of nodes' degrees and A is the adjacency matrix.

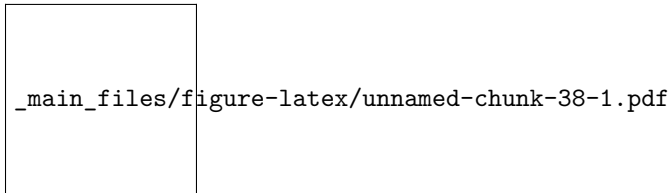
The Laplacian matrix has interesting properties, the best of all being that the number of zeros among its eigenvalues yield the number of components of the graph (one 0 means there is a single connected component, two zeros means the network is split in two disconnected components, etc.). The eigenvectors associated the zero eigenvalues can inform on the membership of nodes to the different components.

The heuristics of the spectral clustering is to use the eigenvectors associated with the set of small non-zero eigenvalues of L to deduce the “almost components”

of the graph. To do so, one chooses a number of no-zero eigenvalues to keep and then the algorithm uses the associated eigenvector to find the groups through a K-means algorithm on the space obtained with these eigenvectors.

Here is an example of spectral clustering on the previous block-model graph.

```
SC<-spectral_clustering(graph_from_adjacency_matrix(sbmnet$networkData),3)
plotMyMatrix(sbmnet$networkData,clustering=list("row"=SC,"col"=SC))
```



9.6.8 Nestedness

The idea of nestedness is to measure the tendency for specialists to only interact with a subsample of the interactors of generalists. In other words, a network would be said “non-nested” only when specialists would interact more likely with specialists than with generalists.

Several indices have been developed over the years. Navigating their complexity is the purpose of the synthesis written by Podani and Schmera (2012).

I will not delve too long on the topic of nestedness because recent research seem to indicate that the concept is not really fruitful for a variety of reasons: - it seems to be very strongly linked to connectance - the diversity of nestedness indices tend to obfuscate its meaning (results reported show more about the index than about the concept) - these different indices can yield very different results when applied to the same dataset - in the classic case of power law distributions of degrees, the nestedness scores tend to be completely determined by the exponent of the power law - finally, when putting the common datasets in the right matrix ensemble (i.e. using correct null models to sample similar matrices), reported nestedness values seem to be completely expected.

9.6.9 Last words

- ecological interaction networks can be used to study community or ecosystem properties
- random network models can help gauge generalities and specificities of networks
- all network patterns do not have a mechanistic model-based explanation yet, but some do (like the link between feasibility, stability, or robustness and connectance)
- interaction network data of different types lend themselves to different analyses (e.g. modularity for binary networks, value adjacency matrix with

signed interactions for May-like analyses)

9.7 Literature

Astegiano, J., Guimarães Jr, P. R., Cheptou, P.-O., Vidal, M. M., Mandai, C. Y., Ashworth, L. & Massol, F. (2015) Persistence of plants and pollinators in the face of habitat loss: Insights from trait-based metacommunity models. *Advances in Ecological Research* - vol. 53 (eds G. Woodward & D. A. Bohan), pp. 201-257. Academic Press.

Bizeul, P. & Najim, J. (2021) Positive solutions for large random linear systems. *Proceedings of the American Mathematical Society*, 149, 2333-2348.

Clauset, A., Shalizi, C. R. & Newman, M. E. J. (2009) Power-law distributions in empirical data. *SIAM Review*, 51, 661-703.

Dunne, J. A., Williams, R. J. & Martinez, N. D. (2002) Food-web structure and network theory: The role of connectance and size. *Proceedings of the National Academy of Sciences*, 99, 12917-12922.

Dunne, J. A., Williams, R. J. & Martinez, N. D. (2002) Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, 5, 558-567.

James, A., Pitchford, J. W. & Plank, M. J. (2012) Disentangling nestedness from models of ecological complexity. *Nature*, 487, 227-230.

Leger, J.-B., Daudin, J.-J. & Vacher, C. (2015) Clustering methods differ in their ability to detect patterns in ecological networks. *Methods in Ecology and Evolution*, 6, 474-481.

Lewinsohn, T. M., Prado, P. I., Jordano, P., Bascompte, J. & Olesen, J. M. (2006) Structure in plant-animal interaction assemblages. *Oikos*, 113, 174-184.

May, R. M. (1972) Will a large complex system be stable? *Nature*, 238, 413-414.

Newman, M. E. J. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103, 8577-8582.

Pocock, M. J. O., Evans, D. M. & Memmott, J. (2012) The robustness and restoration of a network of ecological networks. *Science*, 335, 973-977.

Payrató-Borràs, C., Hernández, L. & Moreno, Y. (2019) Breaking the Spell of Nestedness: The Entropic Origin of Nestedness in Mutualistic Systems. *Physical Review X*, 9, 031024.

Podani, J. & Schmera, D. (2012) A comparative evaluation of pairwise nestedness measures. *Ecography*, 35, 889-900.

Stone, L. (2016) The Google matrix controls the stability of structured ecological and biological networks. *Nature Communications*, 7, 12857.

- Strona, G., Nappo, D., Boccacci, F., Fattorini, S. & San-Miguel-Ayanz, J. (2014) A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature Communications*, 5.
- Stumpf, M. P. H. & Porter, M. A. (2012) Critical truths about power laws. *Science*, 335, 665-666.
- Thomas, M., Verzelen, N., Barbillon, P., Coomes, O. T., Caillon, S., McKey, D., Elias, M., Garine, E., Raimond, C., Dounias, E., Jarvis, D., Wencélius, J., Leclerc, C., Labeyrie, V., Cuong, P. H., Hue, N. T. N., Sthapit, B., Rana, R. B., Barnaud, A., Violon, C., Arias Reyes, L. M., Latournerie Moreno, L., De Santis, P. & Massol, F. (2015) A network-based method to detect patterns of local crop biodiversity: validation at the species and infra-species levels. *Advances in Ecological Research*, 53, 259-320.
- Williams, R. J. & Martinez, N. D. (2000) Simple rules yield complex food webs. *Nature*, 404, 180-183.
- Yang, Z., Algesheimer, R. & Tessone, C. J. (2016) A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, 6, 30750.
- Yodzis, P. (1998) Local trophodynamics and the interaction of marine mammals and fisheries in the Benguela ecosystem. *Journal of Animal Ecology*, 67, 635-658.

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL <http://yihui.org/knitr/>. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL <https://github.com/rstudio/bookdown>. R package version 0.37, <https://pkgs.rstudio.com/bookdown/>.