BIOL 638: Computational Ecology

Spring 2016 Syllabus

Basic information

Instructor: Dr. Gareth J Russell, russell@njit.edu, 973 596 6412

Meeting time: Mondays, 11:30am to 2:30pm

Location: CKB 326, NJIT campus

Office hours: Mondays, 2:30pm to 3:30pm, or by appointment

Pre-requisites: A willingness to embrace computing!

Course description

Computational Ecology is simultaneously an introduction to computational techniques, covering a wide range of topics, and an introduction to mathematical and computational aspects of ecology. Each class covers a computational topic and an ecological topic, with the ecological topic providing the motivation for the computational topic. Because of this…

…you do not need a background in ecology, or to be studying ecology, to benefit from this course!

If your interests are not ecological, you can still learn some useful computational skills and ways of thinking that will almost certainly be applicable in your own field of study. Furthermore, your term project can be on any topic of your choosing.

Learning objectives

1. Understand the breadth of computational techniques employed in the modern scientific enterprise (as exemplifed by ecology).
2. Understand different computing paradigms (procedural, functional, object-oriented, etc.) and their relative merits.
3. Understand the power of range of computing applications (symbolic solving, numerical intregration and simulation, linear algebra, image processing, visualization, data processing, etc.).
4. Be able to translate a concept into an algorithm and then into actual code.
5. Explore in more depth some specific computational techniques that apply to your own interests.
6. Learn some of the mathematical and computational underpinnings of our understanding of the natural world.

Software and texts

We will be using the *Mathematica* software package in this course. We assume that you will be bring your own laptop to class and run *Mathematica* on that. If not, we can make arrangements to use an NJIT machine in class, but you will still need to find a computer for homework.

How do I get *Mathematica*?

There are 3 options:

1. Download the NJIT site license desktop version. Cost: $0. You must be connected to NJIT campus network (by VPN if off-campus) for it to run.
2. Purchase a student license, desktop version. Cost: $140 unlimited, $70/year, $45/semester. No need to be connected to any network.
3. Student license, web version. Cost: $70/year or $10/month. Works on web browser. Must be connected to the internet. Interface speed depends on quality of connection. Notebooks and other files will be stored in the Wolfram Cloud.

Rutgers students

The NJIT site license option is available to you as long as you go through the steps of activating your NJIT account (as are the other buy-it-yourself options, of course). But there is an additional option. If you are currently a TA employed by Rutgers, you are considered ‘staff’, and can download the same software for free from Rutgers’ software portal. In the past, Rutgers has not required a permanent internet connection to run it, so that frees you somewhat. In the past this has not been available to RAs, scholarship holders, etc.

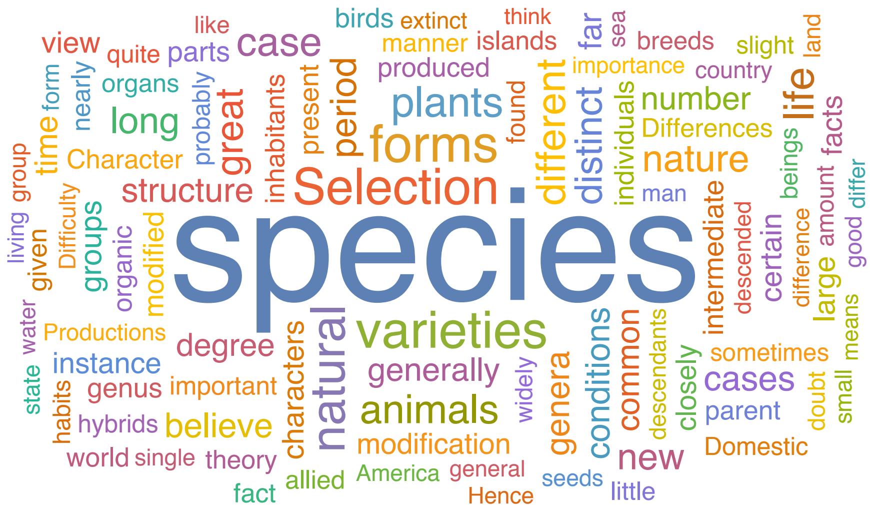
Why *Mathematica?*

Mathematica is a popular software package among mathematicians and physicists, less so among biologists, who tend to use R, MATLAB etc. We use it in this course for a number of reasons.

- It is the software package that I, your instructor, knows the best, and the course materials (see below) were developed in it over many years.
- *Mathematica* has the widest array of abilities of any software package on the market, which means we can also explore a wide range of computational topics without having to switch environments. These abilities are included in the base package — no add-ons to load, no separate sets of functions to learn.
- *Mathematica* uses a *notebook* interface that was, until recently, unique. It has a number of advantages, but is particularly good for teaching, because you everything you will do (code, annotation, graphics, presentation) is held in a single document. *This syllabus is a Mathematica notebook!*
- *Mathematica*’s language is the most internally consistent of any of the high-level languages. By analogy to a regular language, it has a consistent and regular grammar and syntax. This makes these aspects easier to learn.

But importantly, the point of the course is not to learn a specific language, but to learn the principles of computation, and for that we could use any one of a number of software languages. Once you understand the principles, learning another language is not very difficult.

One downside: *Mathematica* is *so* powerful and its vocabulary is *so* large that it can seem overwhelming. Stick to the basics and you will be doing cool things very quickly. (The ‘Origin of Species’ word cloud below is the result of a single (hidden) line of code.)



The course notebooks

There is no set external textbook for the course. Instead, the text is a series of *Mathematica* notebooks that you will download (the link will be given in the first class).

For programming help, *Mathematica*'s help system is vast and comprehensive, and there are any number of online videos to help you (you will follow along with some of these at the beginning). Having said that, the creator of *Mathematica* recently released a sort of 'beginner's guide' book which is $19.95 for students (http://www.wolfram.com/books/profile.cgi?id=9239). You might find it useful.

If you are an ecology student, or want some more background on the ecology component of the course, I *recommend* you get any one of the many good primers of mathematical ecology, such as:

- *An Illustrated Guide to Theoretical Ecology* by Ted Case
- *A Primer of Ecological Theory* by Joan Roughgarden

You may already have one that is suitable.

Assessment

Component	Grade percentage
Homeworks	40
Term paper	50
Participation in class	10

What is participation? In this small, graduate level class, it means being alert in class, asking questions, trying to answer them, and generally not hiding in a corner. It doesn't matter if you are an MS student, a PhD student, or even an undergraduate taking the class by special permission — everyone can, and should contribute. As long as you make that effort, you will get 10. I will let you know quickly if your participation grade starts to drop!

Topic outline

Given that this is a small, graduate-level class, the syllabus, especially towards the second half of the semester, is flexible so that we can explore topics that align with students’ interests. Below is a *typical* progression.

Date	2016−01−25
Class	1.
Computation	Introduction. Mathematica as an environment. Notebooks, document structure and formatting. Executing code. Finding help. Examples of computation in ecological research. How computers store and work with numbers. Symbols and symbolic computation. Mathematical representation vs. computer code. Functions. Nesting. Everything is a function! The importance of 'play.'
Ecology	Ecological units, and how we might code them for computation. Models, assumptions and approximations.
In class	Format the notebook for your class notes. Set it up with a title, headings, etc.
Homework 1	Go through "Hands on Start to Mathematica," following along in a notebook.
Homework 2	Find a data file with numeric or textual data (or both) to bring to the next class. Could be a text file, an Excel document, even a binary file. No pictures.
Turn−in	Turn in your notebook and proposed data file (so I can check it for suitability for the next class).

Date	2016−02−01
Class	2.
Computation	Review of numbers and symbols. Text. Lists and parts of lists. Table vs. Map. Extracting elements and pattern matching. Vectors and matrices. Importing and manipulating data. ListPlots and ArrayPlots.
Ecology	More on ecological units. Models, assumptions and approximations. Common types of ecological data: populations, communities, distributions, interactions. Graphical representations of ecological data.
In class	Import your data, and play with it. Change the way it's coded, extract different subsets, make graphics.
Homework 1	Creating, manipulating and plotting lists and matrices.
Homework 2	
Turn−in	Turn in your class notes and your homework exercise.

Date	2016−02−08
Class	3.
Computation	Continuation of lists. Reformatting data. Functions. Writing your own functions. Continuous functions. Plotting continuous functions. Pure functions. Nesting functions and the "Nest" function.
Ecology	Single−species population models — discrete time. Discrete and continuous individuals. Geometric growth.
In class	Continue manipulation and plotting of your data.
Homework 1	Discrete−time population growth.
Homework 2	Turn in the notebook in which you have explored your data file as well as your homework.
Turn−in	

Date	2016−02−15
Class	4.
Computation	More on nesting. Random numbers. Graphics. The structure of graphics. Functions with options. Interactive graphics with Manipulate. Introduction to calculus. Differential equations. Symbolic vs. numeric solving.
Ecology	Continuous−time population growth. Adding discrete space to population models. Discrete diffusion on a lattice. Types of boundaries.
In class	
Homework 1	Write proposal for term paper.
Homework 2	
Turn−in	Proposal

Date	2016−02−22
Class	5.
Computation	More on differential equations and symbolic vs. numeric solving (integration).
Ecology	Resource limitation. Density dependence. Logistic model in discrete and continuous time. Population cycles and chaos. Demographic and environmental stochasticity. Time−lags.
In class	Write your own function for density−dependent discrete−time growth (the logistic equation). Explore it graphically by using Manipulate.
Homework 1	Finalize term paper topic and begin working on it. Also, find an interesting image or a biological structure (landscape, organism with interesting pattern, leaf, etc.) and bring it to the next class.
Homework 2	
Turn−in	

Date	2016−02−29
Class	6.
Computation	Convolution and cellular automata. Introduction to image manipulation — filters.
Ecology	Diffusion again. Real populations on complex landscapes.
In class	Working with images.
Homework 1	Work on term paper.
Homework 2	
Turn−in	

Date	2016−03−07
Class	7.
Computation	Matrix algebra. Eigenvectors, eigenvalues and stability (part I).
Ecology	Population structure. Life tables, Leslie matrices, elasticity analysis. Stable age distribution.
In class	More with images. Discussion of term paper.
Homework 1	Work on term paper.
Homework 2	
Turn−in	Working draft of term paper.

Date	2016−03−14
Class	8.
Computation	Isoclines, fixed points, vector fields.
Ecology	Harvesting models. Functional responses.
In class	Work on term paper.
Homework 1	Work on term paper.
Homework 2	
Turn−in	

Date	2016−03−21
Class	No Class
Computation	
Ecology	
In class	
Homework 1	
Homework 2	
Turn−in	

Date	2016−03−28
Class	9.
Computation	Interactive interfaces.
Ecology	Two−species models 1. Consumers and resources. Harvesting strategies. Functional responses.
In class	Term paper troubleshooting. Student−led review of concepts.
Homework 1	Work on term paper.
Homework 2	
Turn−in	

Date	2016–04–04
Class	10.
Computation	
Ecology	Two–species models 2. Predator–prey and host–parasitoid systems. Phase space plots, stable and unstable equilibria. Limit cycles.
In class	Work on term paper.
Homework 1	Work on term paper.
Homework 2	
Turn–in	

Date	2016–04–11
Class	11.
Computation	
Ecology	Two–species models and more. Competition. Multiple stable equilibria, saddles. Generalized Lotka–Volterra model. Community stability.
In class	
Homework 1	
Homework 2	
Turn–in	

Date	2016–04–18
Class	12.
Computation	
Ecology	Multiple–species models. Stability vs. complexity debate. Food web structure. Community assembly and succession.
In class	
Homework 1	
Homework 2	
Turn–in	

Date	2016–04–25
Class	13.
Computation	
Ecology	
In class	Presentations of term paper projects.
Homework 1	Finalize projects based on feedback.
Homework 2	
Turn–in	

Date	2016–05–02
Class	14.
Computation	
Ecology	
In class	Presentations of term paper projects.
Homework 1	Finalize projects based on feedback.
Homework 2	
Turn–in	