RoLE Model User Guide

Renata Diaz, for now

4/21/2023

Table of contents

Preface

This is a user guide for working with the RoLE model, which includes the roleR and roleShiny R packages.

Part I

Background

What is the RoLE Model? Who is involved? What are these packages?

1 Process modeling in ecology and evolution

What is process modeling? Why do we use it for eco-evo

2 Models rolled into RoLE

The RoLE model implements versions of a number of established models.

2.1 Ecological neutral theory

2.2 Double-neutral ecological and population genetic drift

2.3 Competitive coexistence

2.4 Lotka-Volterra

2.5 Island biogeography

3 RoLE Mission

3.1 Scientific vision

3.2 Operating principles

3.3 Code of Conduct

Part II

How a RoLE model works

No code, high level narrative (or visuals!) of what happens in a RoLE model.

4 Components of a RoLE model

4.1 Metacommunity

4.1.1 Species, traits, phylogeny, genetics

4.2 Local community

4.2.1 Species, traits, phylogeny, genetics

5 Model inputs: Setting the stage

5.1 Parameter inputs

5.2 Pre-set configurations

6 Timestep-to-timestep

What happens when the model actually runs? To really understand the model, we have to understand its steps and the logic and implications of each one.

Each model run is composed of several timesteps, and during each step events like death, birth, immigration, and speciation happen to individuals or species in the model.

These events happen randomly based on the model parameters, causing the state of the local and meta communities to change as the model is run.

6.1 Deaths

Every timestep, no matter what, an individual in the local community dies.

If we are using a neutral model (parameter neut_delta = 1), a completely random individual is always chosen for death. This is the definition of neutral evolution!

But if the model is not neutral, an individual's chance of death (or ability to survive in its environment) is based on its ability to compete with other individuals and survive in its environment.

The closer the trait of the individual is to the environmental optimum of 1, a perfect match to its environment, the more likely it is to survive.

But at the same time, the more individuals that have similar traits to it, the more the resources each needs will overlap, and this competition will make it harder to survive. (Add more on theory here).

An outcome of RoLE's individual-based model framework is that intraspecific competition is modeled, and even emphasized. Since individuals of the same species typically have similar traits, they will compete more. (could add info on what comp and env sigma do, but this info might belong in parameters section).

6.2 Birth and dispersal

Every timestep, no matter what, EITHER birth or dispersal causes a new individual to appear in the local community.

Birth is when an individual in the local community reproduces to make a single offspring.

Dispersal is when an individual is born in the metacommunity and immigrates from the metacommunity to the local community.

The chance of each is based on the dispersal_prob param, which is simply the chance of dispersal (making 1-dispersal_prob the chance of birth).

6.3 Speciation

Speciation in RoLE works by making the individual that was just birthed or immigrated a new species.

Speciation may or may not happen each timestep, with a chance equal to speciation_local.

Realistically speciation_local is an extremely small value like 0.001 to capture the rarity of speciation events over time.

When speciation happens, a new tip is added to the phylogeny for the new species.

6.4 Trait change

When birth, dispersal, or speciation happens, the new individual gets a new trait deviating from the traits of its parent.

This deviation depends on the trait_sigma parameter - the larger the trait_sigma the more child traits will deviate from parent traits. (this is also based on speciation meta and extinction meta, which could be worth explaining)

7 Model results

What are the results of a role model?

7.1 Raw community state

7.2 Summary statistics

8 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

Part III

Running RoLE models

Ok, how do I actually play with RoLE?

9 Installation

9.1 Compiled binaries

9.2 From source

10 RoLE Shiny

10.1 Installation

10.2 Access

10.3 Using

10.4 Saving results

11 Your first RoLE model

11.1 Model specification

11.2 Running

11.3 Results and interpretation

12 RoLE Experiments

12.1 RoLE Experiments

13 Reproducibility

13.1 Stochasticity and non-repeatability

13.2 Storing model results

14 (Not implemented) RoLE models at scale

14.1 Paralell and cluster computing

Part IV

Use cases

Overview of main (forseen) use cases for RoLE.

15 RoLE Models for In-Silico Exploration

Generate and test hypotheses for how different types of conditions/interventions affect model outcomes.

16 Linking pattern to process in empirical data

16.1 Many-to-one mapping

16.2 Likelihood-free inference

16.3 Worked example

17 RoLE in the Classroom

Tips, tricks, and sample curricula from using RoLE as a teaching tool.

Part V

Troubleshooting

What to do when it doesn't run.

18 Contact

We respond to GitHub issues!

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.