

# Simplex: An individual-based platform for ecological modeling

*Kenneth J. Locey*

Department of Biology, University of Indiana, Bloomington

email: [kjlocey@indiana.edu](mailto:kjlocey@indiana.edu)

## ABSTRACT

A popular approach in ecological modeling explicitly simulates the biology and ecology of individual organisms. Individual-based models (IBM) allow ecological patterns and dynamics to be studied as they emerge from the individual level according to simulated principles and theories. IBMs also allow virtually any aspect of a simulated system to be tracked and recorded for later analysis. I developed a novel IBM platform (i.e., Simplex) to explicitly simulate physiology, life history, resource-limited growth, fluctuating environmental conditions, and community assembly and structure in both fluid and non-fluid environments. Simplex accomplishes three primary tasks in keeping with its namesake (i.e., a generalized notion of a triangle to arbitrary dimensions). First, Simplex assembles and runs stochastic ecological models from random combinations of over 20 state variables. Second, Simplex stores the output of simulations, and generates and stores animations. Third, Simplex provides R Markdown files for analyzing simulation data. The Simplex source code is freely available, includes a standard ODD description protocol, is open to community-based revision, includes unit testing and time testing, and has been developed to run on modest hardware.

# INTRODUCTION

Modeling is a elementary approach to understanding ecological systems, the influence of ecological processes, and the predictability of ecological patterns and dynamics. Before modern computing, ecological models were almost exclusively equation-based representations of highly simplified systems (Black and McKane 2012, Otto and Troy 2007, Grant and Swannack 2008). Since the advent of personal computing, ecological models have been increasingly constructed to handle greater complexity and to explicitly simulate ecological processes, forces, and mechanisms.

Simulation-based models are often used to examine analytically challenging or intractable scenarios. Examples are markov models that simulate stochastic demographic changes and early ecological null models, both of which operate on matrices (Gotelli and Entsminger 2001, Hubbell 2001). Ecologists have also simulated growth and interactions among individual organisms for over two decades with individual-based models (IBMs) (DeAngelis and Gross 1992, Rosindell et al. 2015). IBMs explicitly encode rules of how individuals change and interact. Once these rules are encoded, population to ecosystem-level dynamics can emerge as an IBM simulates over time and spatially explicit environments.

A body of literature including original research, comprehensive reviews, and textbooks reveal the use, advantages, and challenges of ecological IBMs (e.g., Grimm 1999, Grimm and Railsback 2005). IBMs can provide degrees of ecological realism, individual variability, and spatial heterogeneity that are unattainable with other models. IBMs also offer the potential for realistic and unanticipated ecological dynamics and patterns to emerge from individual level interactions. However, IBMs come with challenges that include greater computational complexity, the difficulty of explicitly encoding ecological theory, and the challenge of using IBMs to solve applied problems (Matthews et al. 2007, Grimm and Railsback 2005).

Despite their frequent use, the power of ecological IBMs has yet been leveraged to its greatest advantage. Few, if any, include fluid dynamics or combine fluid dynamics with

growth and dispersal. While growth, sensing, and decision making are often modeled, few IBMs integrate physiology, nutrient-limited growth, evolution, environmental flux, community ecology, biogeography, and stochastic sampling theory. Instead, ecological IBMs are more often constructed to model a specific system than to synthesize general theories of ecology and evolution (but see Rosindell et al. 2015). Yet, IBMs can allow researchers to creatively and explicitly simulate any number of theories and principles. Likewise, IBMs can be made to track and analyze any information from physiological changes and population demography, to community assembly and the structure of entire ecosystems.

Here, I present a description of an ecological IBM platform I refer to as “Simplex”. This platform simulates a broad range of ecological conditions, records information from individuals to the ecosystem level, and provides R computing code for analyzing output. Simplex incorporates fluid and non-fluid dynamics with stochastic and selection-based changes within and among populations. Simplex also integrates nutrient-limited growth and physiology. I constructed Simplex with the aim of simulating ecologically complex scenarios, from which the results could be used to test (i) predictions of ecological theory and (ii) the general influence of constraints and processes on populations, communities, and ecosystems.

## METHODS

### Model description following ODD Protocol

The overview, design concepts, and details protocol (ODD) is a standard for describing individual-based models (Grimm et al. 2006). Here, I provide an abbreviated version of the Simplex ODD protocol. The full Simplex ODD protocol can be found on the public GitHub repository at <https://github.com/LennonLab/simplex>. Likewise, to save space, I refer the reader to the README.md file on the public Simplex GitHub.com repository for descriptions of source code, files, and directories (<https://github.com/LennonLab/simplex/blob/master/README.md>).

### Purpose

The purpose of Simplex is to explicitly simulate life history, the evolution of traits, the assembly of ecological communities, and the structure and function of ecosystems in a spatially explicit environment under stochastic conditions, environmental filtering, and species sorting. The proximate goal of Simplex is to generate high degrees of variation in the assembly and structure of populations and communities by assembling many different models from random combinations of state-variables and processes. The ultimate goal of Simplex is to provide a simulation-based platform for examining conditions under which processes and constraints have a robust influence on eco-evolutionary dynamics, patterns of biodiversity, and ecosystem processes.

### Entities & their state variables

*Individual organisms*—These are distinguished by the corresponding elements of several lists. For example, in simulating reproduction, a Simplex model chooses an individual at random from the list of cell quotas, where the probability of reproducing is determined by the levels of endogenous resources, i.e., cell quota. The first position in the list of cell quotas, and in all other lists of individual attributes, corresponds to the same individual, i.e.,:

```
IndIDs = [1, 2, 33, 14]
Quotas = [0.1, 0.99, 0.14, 0.05]
Xpos = [45, 23, 456, 1]
Ypos = [765, 87, 21, 34]
```

Above, the individual with the ID “1” has a cell quota of 0.1 and is located at position x=45, y=765. Note that in simplex, there are at least three resource quotas. In future versions, the quotas will more explicitly represent nitrogen, carbon, and phosphorus. Of course, each individual could also have been distinguished as its own list of attributes, e.g., Ind1 = [1, 0.1, 45, 765]. However, this would require a large lists of lists and could make quantifying some metrics (e.g., generating a frequency distribution of cell quotas, ages, or environmental optima) more computationally intensive than necessary.

Lists of attributes for individual organisms include the time each individual spends in the system, species ID, individual ID, 2D spatial location, endogenous levels of resources (resource specific cell quotas), individual-level metabolic maintenance cost, individual-level maximum dispersal rate, parental lineage, and individual resource use efficiency for each resource.

*Species*—Each species is characterized by the individuals that share a common set of traits, such as maximum growth rate, maximum dispersal rate, and metabolic maintenance cost. Species information is stored in dictionaries. In this way, if Simplex requires the species ID of an individual it will access the species ID list. But, if Simplex requires the maximum growth rate for an individual, it will first find the species ID and then use that ID to access the dictionary for maximum specific growth rates of species. Species-level data include metabolic maintenance cost, maximum dispersal rate, maximum theoretical growth rate, environmental optima, and the resource use efficiency for each resource.

```
IndIDs = [1, 2, 33, 14]
spIDs = [12, 32, 11, 6]
MaxGrowthDict = {12: 0.9, 32: 0.5, 11: 0.8, 6: 0.4}
```

The individual with ID of 1 belongs to species 12, which has a theoretical maximum growth rate of 0.9. By storing species-level information in dictionaries, Simplex does not have to store redundant information in lists.

*Resource particles*—Like individual organisms, individual resource particles distinguished by the corresponding elements of several lists. Each list holds a specific type of information, including time spent in the system, particle ID, 2D spatial location, the type of resource particle, the size of the resource particle.

```
resIDs = [4, 6, 17, 1]
size = [100, 87, 156.3, 0.001]
```

```
Xpos = [56, 34, 567, 2]
Ypos = [876, 98, 32, 45]
```

Above, the particle with ID of “4” has a size of 100 and is located at position x=56, y=876.

*Inert tracers particles*—These are objects that move or flow into and through the environment. Inert tracers only interact with physical barriers. The use of these tracers particles allows for physically dynamic attributes of a system to be quantified. Examples are ecosystem residence time and the ease of passive movement or inertia through the system. Like other individual-level objects in Simplex, tracers are distinguished by corresponding elements of several lists. Each specific position in the list corresponds to the same tracer particle. Information stored about each tracer particle includes time spent in the system, 2D spatial location and particle ID.

*Physical barriers*—These objects are simulated as discrete 2D spatial coordinates that cannot be occupied by any individual entities. The number, size, and location of physical barriers are chosen at random by Simplex at the start of each model.

### System level state variables

Each run of Simplex begins with random choices for the specific structure of a log-series metacommunity and values of width (5 to 100), height (5 to 100), flow rate (0.0 to 1.0 units along the x and y axis per time step), the number, size, and location of physical barriers, the number and direction of environmental gradients, the rate and severity of stochastic disturbance, the rate of fluctuation in basal flow rate, and degrees of fluctuation and synchronization for the inflow of individual organisms and resources, which range from completely in-sync to completely out of sync.

### Spatial and temporal scales

The two general aspects of scale are grain (i.e., resolution) and extent (e.g., total area). The environment of Simplex models is two dimensional and can vary along each axis from 5 to

100 discrete units. This makes for a potential total extent of 25 to 10,000 discrete patches, each with a grain of 1 square unit. However, all particles move in decimal units, meaning that individual particles can occupy practically infinite locations within patches. Extent of time in Simplex models refers to residence time, i.e., the average amount of time that individual particles spend in the system. As shown by test simulations, residence time for individual particles (organisms, resources, inert tracers) can vary across five orders of magnitude.

Grain is the smallest unit over which change can happen. For example, the original ODD documentation states “One time step represents one year and simulations were run for 100 years. One grid cell represents 1 ha and the model landscape comprised 1,000 x 1,000 ha”. In Simplex, spatial grain is effectively determined by slowest rate at which individuals can traverse the landscape, which can vary from 0.00001% to 10% of the x or y axis in one time step.

## Process overview and scheduling

*Assembly.* The user runs a program that chooses random values for system-level state variables including whether disturbance, immigration, speciation, fluid dynamics, etc. will occur and at what rates.

*Core simulation process.* Simplex begin simulating immediately after assembling a model from random combinations of state-variables and processes. Simplex models advance turnover of the environmental matrix according to the rate of flow, which can be as low as 0.0 (i.e., an effectively static environment) or as high as 1.0 (i.e., where inert particles will flow 1.0 units of distance unless hindered by barriers). After each iteration of flow, every individual is given the chance to consume, grow, reproduce, starve, die, and to disperse towards environmental optima. However, each of these processes is probabilistic.

*Simulated life history.* Simplex models simulate growth, reproduction, and death via weighted random sampling. This simulates the partly probabilistic and partly deterministic nature of environmental filtering and individual-level interactions. For example, individuals with a



high cell quota have a low probability of dying and a high probability of reproducing.

**Inflow/Entrance.** Resources and individuals enter from any point in the environment. Species identities of inflowing propagules are chosen at random from a log-series distribution, which often approximates the distribution of abundance among species in ecological communities (Hubbell 2001). Along with a species ID and species-specific maximum rates of resource uptake, active dispersal, resource efficiencies, cell maintenance, and environmental optima, each propagule was given a unique ID and a multi-resource cell quota that represent the state of internal resources. The average of these cell quotas determines the probability of reproduction.

**Dispersal.** Individuals are allowed to actively move along environmental gradients and even against the direction of environmental flow and towards their environmental optima. A better match to the environmental optima increases the chance of reproduction and an individual's ability to consume, grow, and reproduce.

**Consumption & growth.** Sampled individuals consume resources according to their specific maximum rates of uptake and grow according to specific resource efficiencies and maintenance costs. However, consumption can only happen in Simplex if individuals physically encounter resource particles. Uptake increases the individual cell quotas and decreases the size of the resource particle. Individual cell quotas are then decreased according to a specific physiological maintenance cost.

**Reproduction.** Reproduction in Simplex is currently limited to clonal reproduction with the possibility of mutation. Individuals reproduce with a probability determined by the combination of mean cell quota and the proportional match to the environmental optima. The cell quota of each resource is evenly divided between the individual and its daughter. The daughter is then given a unique individual ID and the species ID of its mother, i.e., unless in the case of a random speciation event.

**Speciation.** Speciation is simulated within Simplex as a discrete event (*sensu* Hubbell

2001 and Bell 2001) and is accompanied by mutations in the values of species-level state variables, much like that expected from a Wright-Fisher model (Hartl and Clark 1997). These random events allows for diversity to arise within the system, which the environmental filter can then select on.

**Death and emmigration.** Individuals sampled at random will die if their smallest resource specific cell quota (i.e.,  $N$ ,  $C$ ,  $P$ ) is equal to or less than 0. Individuals, resource particles, and inert tracers are considered to have left or to have flowed out when they pass beyond edges of the environment.

*Fluid dynamics.* Simplex uses an efficient and powerful method for simulating fluid flow, i.e., a Lattice-Boltzmann Method (LBM) (Succi 2001). An LBM discretizes the environment into a lattice and attaches to each position in the lattice a number of particle densities and velocities for each simulated direction. Simplex simulates nine directions of movement possible in a 2D spatial environment (i.e., N, S, E, W, NE, NW, SE, SW, current position).

*Active dispersal.* Simplex models allow individuals to move towards their environmental optima. Rather than a single environmental optima resulting from a single environmental gradient, Simplex allows environmental optima to occur as intersections among environmental gradients. Hence, individuals potentially have multiple optima resulting from unique and equally optimal intersection of up to 10 environmental gradients.

*Duration: A run to mean reversion.* Once assembled, a Simplex model simulates ecological processes (birth, death, dispersal, growth, consumption, etc.) until the system reaches a point of mean reversion, i.e., the tendency of a system to reverse a directional change in, say, total abundance. Mean reversion captures whether a system is fluctuating around a mean value. Once 100 generations have occurred within a model, Simplex examines whether a point of mean reversion has occurred by conducting an Augmented Dickey-Fuller (ADF) Test, which is well-explained here: <https://www.quantstart.com/articles/Basics-of-Statistical-Mean-Reversion-Testing>. A model is stopped once mean reversion is

determined to occurred and, unless the number of desired simulations (i.e. models) has been reached, Simplex simply constructs another model from random combinations of state-variables and processes, and then runs it to mean reversion.

## Design concepts

*Ecological complexity.* Simplex assembles models from random combinations of constraints (state-variables) and processes to generate output data that allow the user to test the general influence of state-variables and processes.

*Process-based stochasticity.* Simplex was designed to incorporate random sampling and probability-based changes in as many ways as possible. This does not imply neutral dynamics or completely random dynamics, but instead makes all processes and dynamics (even strong environmental filtering) subject to a degree of uncertainty.

*Nutrient limited growth.* All models assembled by Simplex employ the universal concept that individual growth, reproduction, and activity is limited by resources within individuals (endogenous) rather than resources simply within the environment (exogenous).

## Theories

*Constraint-based theory.* Simplex was originally built to explore the influence of ecosystem residence time (volume/flow rate) on community assembly and structure. That is, the idea that both ecological processes and physical constraints shape ecological diversity.

*Chemostat theory.* Simplex operates much like an unhinged bioreactor or chemostat. That is, particles flow through a system of a defined size at an average rate, and are limited in growth by their residence time.

*Ecological neutral theory.* Simplex operates via random sampling and can vary from being completely neutral (all individuals having equal vital rates) to completely idiosyncratic (all individual and species are as different as possible). The one aspect of neutral theory that Simplex adopts without question is the importance of stochastic life history processes (i.e. weighted or unweighted random fluctuations in population sizes).

## Adaptation

Individuals can move towards their environmental optima. Populations can become aggregated in areas that provide favorable intersections of species optima. Species can evolve by the action of the environmental filter on subpopulation variation in state variables. Individual can seek conditions that match them best to their environment.

## Sensing

Individuals can move towards environmental optima against the flow of the environment. Otherwise, all encounters are the result of random walks or fluid flow.

## Interaction

At the moment, individuals only interact indirectly through excluding each other from resources (e.g. preemption).

## Stochasticity

The occurrence of nearly all processes of birth, death, life, immigration, dispersal, emigration, consumption, etc. are conducted via random sampling. In this way, population and community dynamics result, in part, from demographic stochasticity. Likewise, the emergence of life history traits proceeds from initially random combinations of traits.

## Observation

The following is recorded for each Simplex model:

- Values of randomly chosen state variables
- Total abundance ( $N$ ) and Species richness ( $S$ )
- mean growth rate (per species & per capita) and mean maintenance cost (per species & per capita)
- mean resource efficiency (per species & per capita) and mean active dispersal rate (per species & per capita)
- Compositional turnover: Bray-Curtis, Sorensen's
- Whittaker's species turnover

- Species evenness: Smith and Wilson's index ( $E_{var}$ ), Simpson's index ( $E_{1/D}$ )
- Species diversity: Shannon's diversity ( $H'$ ), Simpson's diversity ( $D_{1/D}$ )
- Dominance: Absolute ( $N_{max}$ ), Berger-Parker index,  $N_{max}/N$
- Productivity in numbers of individuals and in amount of cell quotas
- Residence time:
  - Ecosystem,  $(length * width) / flowrate$
  - Individual, avg time spent in the system before dying or emigrating
  - Resource, avg time spent in the system before washing out or being consumed
  - Tracer, avg time spent in the system before washing out
- Individual residence time distribution (RTD)
- Resource particle RTD
- Tracer particle RTD

These data are stored in .csv files. The file *SimData.csv* is stored as an R-formatted data.frames. These files can be directly imported into an R or Python environment.

## Initialization

The model initiates with a random set of values for state-variables, 100 to 10,000 randomly drawn individuals from a theoretical log-series metacommunity. These values are saved, so that a Simplex model could be programmed to replicate an analysis.

## Input data

Simplex models require no input data, but it might be cool to use an api to grab environmental data or other data from a website to parameterize a Simplex model.

## Submodels & Equations

*Cell quota model of Droop.* In Simplex models, individuals grow according to their amounts of endogenous resources (cell quota).

Droop (1968, 1983) gave a relationship between specific growth rate ( $\mu$ ) and cell quota ( $Q$ ):

$$\mu = \mu'_m(1 - k_q/Q)$$

where  $k_q$  is the minimum cell quota needed for life, also referred to as the subsistence quota.  $\mu'_m$  is the theoretical maximum growth rate.

*Maintenance cost of Pirt.* Pirt (1965) states “The variation, with growth rate, of the yield of organism from the substrate used as energy source is attributed to consumption of energy at a constant rate for cell maintenance.” He derived a relationship between the growth yield (biomass), the growth rate, and metabolic maintenance. Simplex models use Pirt’s concept of a constant maintenance requirement. Simplex also draws from Pirt’s simple relation for substrate use:  $use(total) = use(maintenance) + use(growth)$

Note that respiration and activity without growth is not accounted for.

*Log-series metacommunity.* Simplex models draw immigrating individuals from a theoretical log-series distribution. Hubbell (2001) states that the regional community (i.e., metacommunity) often has a log-series species abundance distribution. The probability density function (pdf) of the log-series is:

$$f(k) = -1/\ln(1 - p) * p^k/k$$

### Notes on Simplex source code

Simplex models operate primarily on lists in a programmatic way, e.g., quickly sorting lists, and removing and returning an element from lists with very little overhead. Likewise, Simplex models generate and hold a lot of information about all the particles and elements in the system, which can become a computationally intensive task. To this end, Simplex modeling coded is written in Python, an easy to read high-level programming language that has many scientific, plotting, and animation libraries. Python gives greater control over the operating

system than data analysis languages (e.g. R, Matlab) that can be comparatively slow at purely computational tasks and can greatly limit the amount of memory held in any data object, and even fail to import large amounts of data. Python can also obtain C-like speeds when implementing certain software, e.g., Cython, Sage.

The output of Simplex is a broad array of information held in seven .csv files. The most important of these is SimData.csv, and is intended to be analyzed in the freely available R (<https://www.r-project.org/>) and RStudio (<https://www.rstudio.com/>) environments. The R statistical computing language is well-suited to the analysis of Simplex output and contains many packages for multivariate analysis and higher-order statistical analysis that Python is only beginning to accumulate. Consequently, I provide R source code in .R files and .Rmd (RMarkdown) files, complete with basic and advanced statistical analyses for analyzing diversity, regression models, ordination, variance partitioning, and for generating pdf documents (via Knitr) that integrate prose, code, and figures for manuscripts.

The reader can view example R-based analysis files that users can use to examine Simplex's simulated data: <https://github.com/LennonLab/simplex/tree/master/results/analyses>.

## Speed & Memory

Simplex models do not complete until the time series of total abundance values reaches a state of mean reversion (i.e., stationarity). Because Simplex models can range from quickly flowing systems with high disturbance to barely flowing and highly stable but depleted systems, simulations can potentially take several minutes or more to complete. Likewise, the ability to simulate many complex scenarios also allows for very large total abundances, the values of which cannot be predicted *a priori* and can even potentially outstrip a computer's memory.

I ran Simplex on a Mid 2010 MacBook Pro (OS X 10.9.5) with a 2.4 GHz Intel Core 2 Duo processor and 4GB of Memory. This system probably represents a below average

capacity for modern personal computers, which for this study, was desirable as Simplex should be able to be ran on both personal computers and high capacity remote servers. I present results for time to completion and required memory in the Results.

## Animations

Simplex can generate animations of its models and store them in various image file formats. It does this using the matplotlib animation library. At the moment, choosing whether to animate or not animate a model is done by commenting out lines of code in model.py. However, this feature will be developed more highly for future versions.

## RESULTS

### Unit tests

Simplex passed all units tests for 15 diversity indices, ensuring that each index returns either the correct calculated value, or 'NaN' if given any values that cannot be used (e.g., negative numbers, string characters, empty lists).

### Speed & Memory

The following are results from 100 randomly assembled models. On average, Simplex models required 96.60 +- 32.70 seconds to run, had an average total abundance ( $N$ ) of 31,367 +- 879, and required 161 +- 4.3 megabytes of memory. The longest any model took to complete was 49 minutes. This particularly demanding model required 261MB of memory and also had generated the greatest total abundance ( $N = 68,808$ ). The shortest any model took to complete was 5.8 seconds with a total abundance of 0 individuals. The least amount of memory any model required was 90MB. These and other analyses from the 100 randomly assembled models can be run using the SpeedMemory.Rmd file located in the results/TestResults directory.

### Products

Simplex generates six files as its output data. They are:

*SimData.csv*. Formatted as an R data frame, where each row is a run of a randomly assembled



model, and each column holds a piece of data about the system that was modeled (e.g., flow rate, total abundance, species richness, species turnover, rate of disturbance, etc.).

*IndRTD.csv*, *ResRTD.csv*, *TracerRTD.csv*. Each line of these three files contains the amount of time that each individual, resource particle, or inert tracer spent in the system. Each line represents the same run of a randomly assembled model.

*RADs.csv*. Each line is a run of a randomly assembled model and contains the rank-abundance vector at the point when the model was stopped.

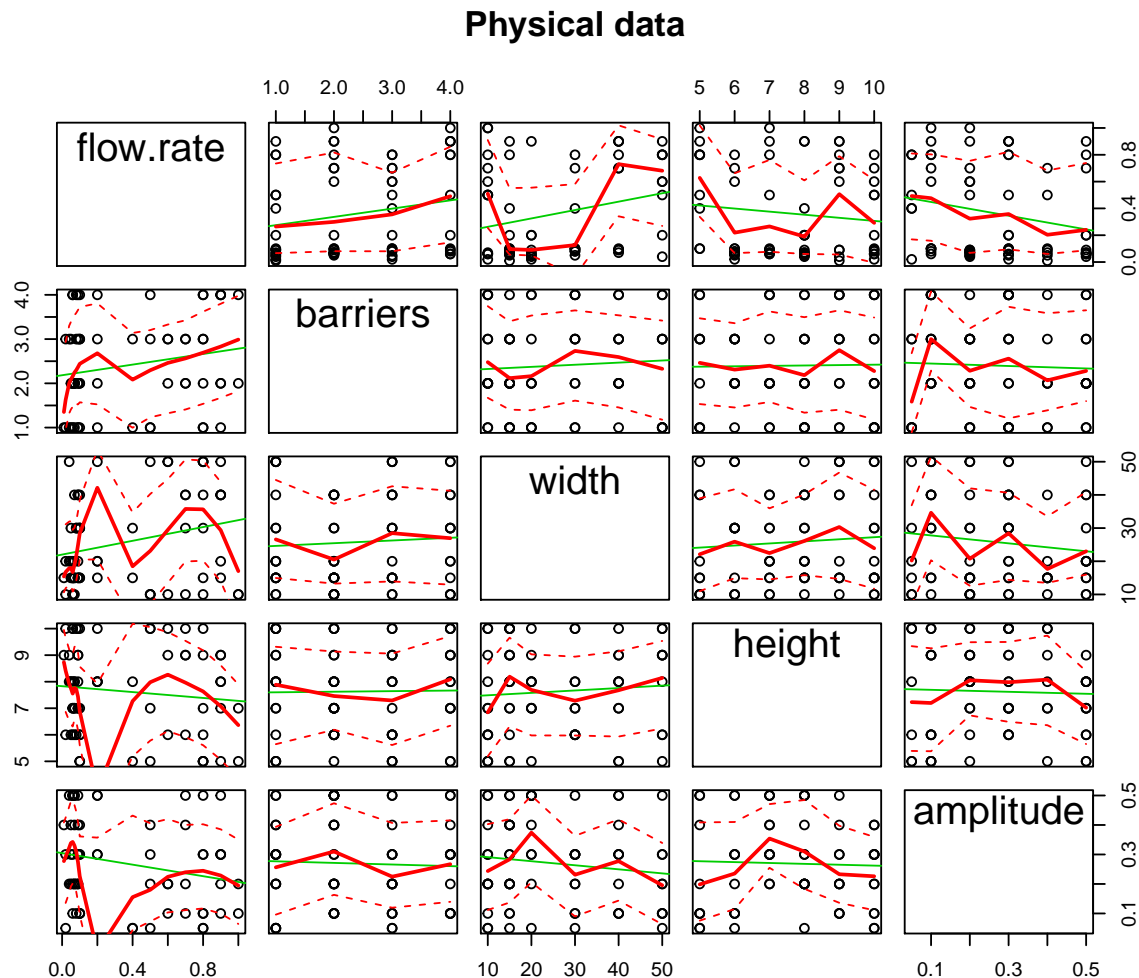
*Species.csv*. Each line is a run of a randomly assembled model and contains a vector of species labels corresponding to RADs.csv.

All output data were correctly formatted and placed within results/simulated\_data/examples directory. Each of the six output data files was able to be imported into the RStudio environment using the Exploratory.Rmd Rmarkdown file provided in the “GitHub/simplex/results/analyses/Rmd/” path. The user can use the Exploratory.Rmd file to craft a .Rmd file for their own specific analyses.

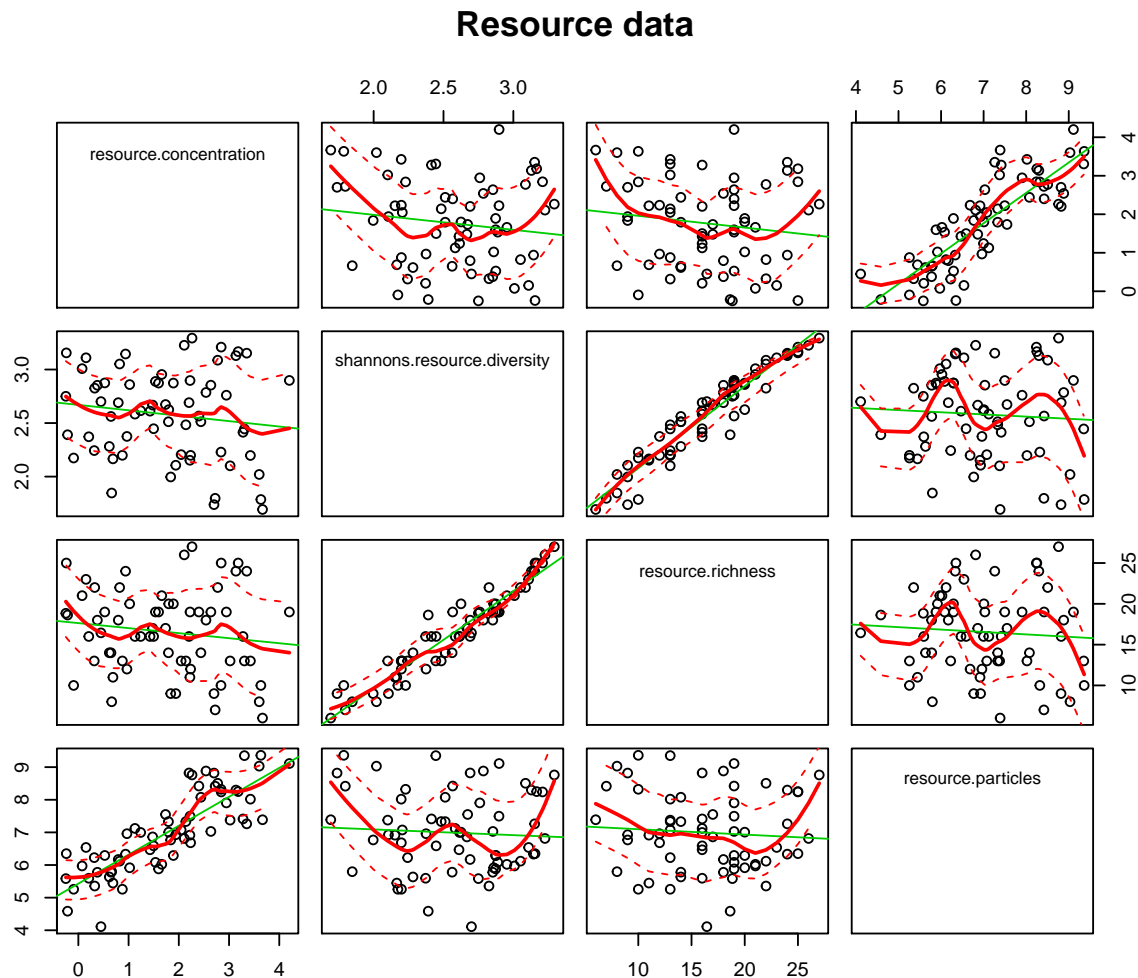
## R analysis files

There are currently two R Markdown analysis files associated with Simplex. These have been included as supplemental files. These are located in the tools/analyses/Rmd directory and are designed to help users understand how to import Simplex files and use R Markdown to analyze Simplex data. The first file, BasicAnalysis.Rmd, focuses on descriptive statistics, multiple univariate analyses, and variance partitioning to reveal relationships among Simplex variables. Running this file on example Simplex data revealed that Simplex produces well-known and/or intuitive relationships (Figs 1, 2, 3, 4).

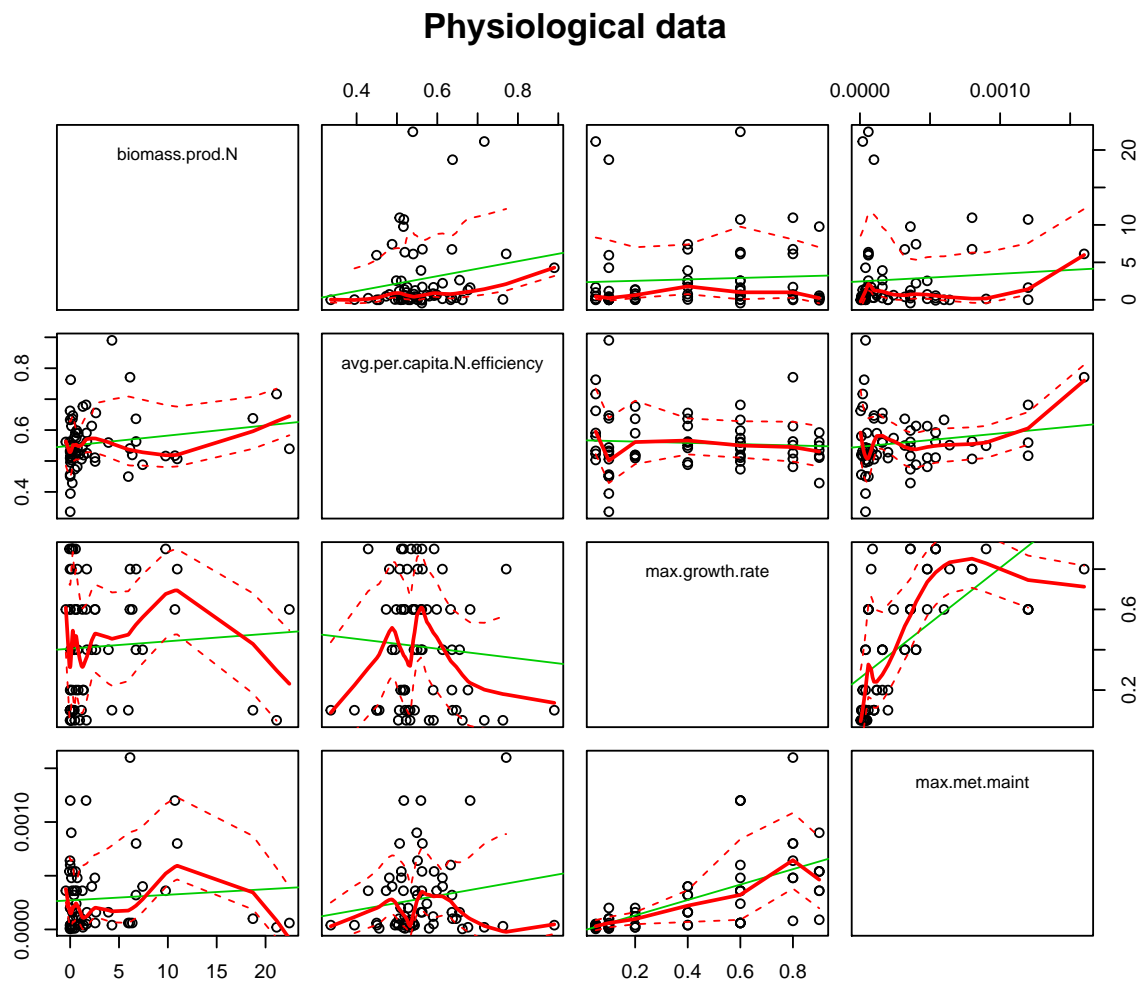
**Figure 1.** A scatter plot matrix revealing that the values of inputs representing physical variables are independent of one another, as should be expected in the modeling. This is only a subset of the physical variables in Simplex. Amplitude pertains to magnitude of temporal fluctuation in the inflow of propagules and/or resources. Barriers pertains to the number of barriers in the system, which varied between 1 and 4.



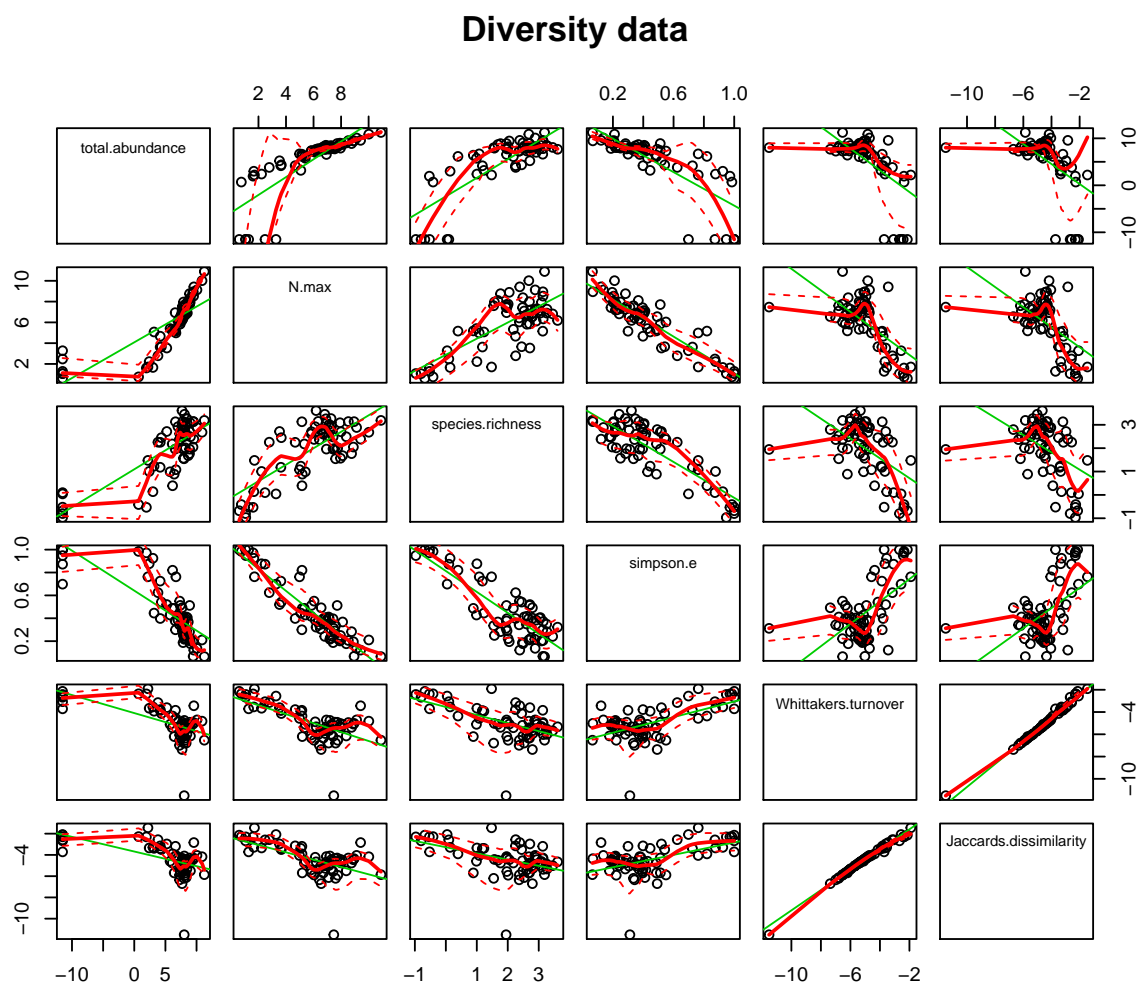
**Figure 2.** A scatter plot matrix revealing that the values of resource-related variables. This is only a subset of resource variables in Simplex. Number of resource particles, their concentration, resource richness, and resource diversity (a combination of resource richness and the variance in abundance among resources) are all highly and positively correlated, as should generally be expected.



**Figure 3.** A scatter plot matrix revealing that the values of physiological variables. This is only a subset of those in simplex. As of 29 October 2015, Simplex does not include explicit stoichiometry. An intuitive life history trade-off arises in that increasing growth rate generally leads to increasing metabolic maintenance.



**Figure 4.** A scatter plot matrix revealing that the values of diversity-related variables. This is only a subset of those in Simplex. There are strong positive relationships between total numbers of individuals (total abundance) and total numbers of species (species richness) and between total abundance and the abundance of the most abundant species (Nmax). Each of these relationships is widely observed in ecological communities (Locey and Lennon 2015). Likewise, different measures of species turnover largely reflect one another, as would generally be expected.



The second file, `Diversity.Rmd`, focuses on the analysis of taxonomic diversity and abundance to include fitting of five common species abundance models using the R-based `vegan` module. `Diversity.Rmd` also provides detailed explanation of each of the five models. Each file generates a knitted pdf (includes code, text, and figures). Pdf files of the output of `Diversity.Rmd` and `BasicAnalysis.Rmd` can be found in the `tools/analyses/Rmd/pdfs` directory.

## Animations

I used the animation module in the matplotlib plotting library to generate avi formatted movies of Simplex models. These can be viewed in any internet browser (<https://github.com/LennonLab/simplex/tree/master/results/movies/examples>). Currently, generating movies requires following three easy steps explained in `model/model.py`.

## DISCUSSION

The Simplex platform assembles individual-based models (IBMs) from random combinations of state-variables and processes, to simulate high degrees of variation and complexity. In this way, Simplex allows ecologists to examine whether the influence of particular variables should be robust to a wide range of conditions. For example, if a user wants to examine the influence of area on commonness and rarity, or the potential relationship between growth rate and physiological maintenance, then the user only needs to run Simplex, allow it to assemble and simulate a large number of models, and then analyze the output files. This is unlike many IBMs that are often built for a specific system or for individuals with highly specified behaviors and abilities (e.g. learning).

In addition to its unique modeling aspects, Simplex includes analysis code that shows users how to import and analyze simulated datasets, including simple univariate relationships, descriptive statistics, variance partitioning, and the fitting of species abundance models. By taking advantage of modern computational tools, version control, social coding, Simplex is well-positioned to continue development as both a research and education/instructional tool for ecology.

Future developments and released versions of Simplex will have trophic dynamics including predator-prey, symbionts and mutualists, and producers and consumers. Individuals will also be attributed with additional behaviors (e.g., search patterns), sensing (e.g., of resources), and learning (e.g., predator avoidance). Additionally, individuals will also be attributed with genomes. Initially, these genomes will be represented by sets of sequences of A's, C's, T's, and G's representing genes that can be mutated as they're passed on through clonal reproduction, or mutated and recombined in the case of sexual haploid or diploid reproduction. A precursor to this has already been developed in Locey (2013), the code for which is available here: <https://github.com/klocey/SYMENT>.

Future developments will also include explicit stoichiometry, where individual resource particles have ratios of Carbon, Nitrogen, and Phosphorus, as well as degrees of biocomplexity. For example, given the chemical structure of phosphate and the size of a resource particle, the particle can be assigned  $x$  units of phosphorus as well as the biocomplexity value of phosphate, which can be estimated as a form of entropy (Bertz 1983). Likewise, improvements and future versions of Simplex will provide increasing numbers of files for statistical analysis of Simplex output. These files will, in a sense, be able to be used as a complete analysis of some particular aspect of Simplex output (e.g., demographic, resource-related, spatial, diversity-related, multivariate, ecological model testing). These and all Simplex files are freely available and open source, allowing users to modify Simplex source code to suit their particular needs.

## ACKNOWLEDGMENTS

I thank Jay T. Lennon, Mario Muscarella, Xiao Xiao, and Ed Hall for fruitful discussions. I thank Dr. Daniel V. Schroeder for his web-based fluid dynamic simulator, the python-based code for which serves as the core method in Simplex for simulating fluid flow. This work was supported by a National Science Foundation Dimensions of Biodiversity Grant (#1442246) awarded to Jay T. Lennon and Ken Locey.

## REFERENCES

- Black AJ, McKane AJ (2012) Stochastic formulation of ecological models and their applications. *Trends in Ecology and Evolution*. 27:337-345.
- Bertz, S (1983) On the complexity of graphs and molecules. *Bulletin of Mathematical Biology* 45:849–855.
- DeAngelis DL, Gross LJ (1992) *Individual-based Models and Approaches in Ecology: Populations, Communities, and Ecosystems*. Chapman & Hall. New York.
- Droop MR (1983) Twenty-five years of algal growth kinetics, a personal view. *Botanica Marina*. 26: 99-112.
- Gotelli NJ, Entsminger GL (2001) Swap and fill algorithms in null model analysis: rethinking the knight's tour. *Oecologia*. 129:281–291
- Grimm V, Railsback SF (2005) *Individual-based Modeling and Ecology*. Princeton University Press, Princeton.
- Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, Giske J, Goss-Custard J, Grand T, Heinz SK, Huse G, Huth A, Jepsen JU, Jørgensen C, Mooij WM, Müller B, Pe'er G, Pioub C, Railsback SF, Robbins AM, Robbins MM, Rossmanith E, Rüger N, Strand E, Souissim S, Stillmane RA, Vabø R, Visser U, DeAngelis DL (2006) A standard protocol for describing individual-based and agent-based models. *Ecological Modeling*. 198: 115-126.
- Hartl DL, Clark AG (1997) *Principle of Population Genetics*. Sinauer Associates, Inc. Publishers, Sunderland, MA.
- Hubbell SP (2001) *The unified neutral theory of biodiversity and biogeography*. Princeton University Press, Princeton, NJ.
- Leadbeater BSC (2006) The 'Droop equation' - Michael Droop and the legacy of the 'Cell-quota model' of phytoplankton growth. *Protist*. 157: 345-358.



Locey K (2013) Synthesized ecological molecular neutral theory. figshare. <http://dx.doi.org/10.6084/m9.figshare.640105>.

Locey KJ, Lennon JT. (2015) Scaling laws predict global microbial diversity. PeerJ PrePrints 3:e1808 <https://dx.doi.org/10.7287/peerj.preprints.1451v1>

Matthews RB, Gilbert NG, Roach A, Polhill JG, Gotts NM (2007) Agent-based land-use models: a review of applications. *Landscape Ecology*. 22: 1447-1459.

Pirt SJ (1965) The Maintenance Energy of Bacteria in Growing Cultures. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 163:224-231.

Rosindell J, Harmon LJ, Etienne RS (2015) Unifying ecology and macroevolution with individual-based theory. *Ecology Letters*. 18: 472-482.

Succi S (2001) *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*. Oxford University Press. New York.