GeneDrive.jl: A Julian Approach to Simulating Biological Dynamics and Control

Valeri Vasquez
Murat Arcak
David Anthoff

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-49
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-49.html

May 1, 2023

GeneDrive.jl: A Julian Approach to Simulating Biological Dynamics and Control

by Váleri Nicol Vásquez

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Murat Arcak
Research Advisor

5/12/2022

* * * * * * *

Professor David Anthoff
Second Reader

5/12/2022

0. Abstract

I. This thesis describes the design, functionalities, and underlying mathematics of GeneDrive.jl, a software library developed to study the effect of biotic and abiotic interactions on biological systems and optimize their control. Named in honor of a new technological horizon in genetic-based biocontrol, the tool is nonetheless broadly applicable to investigations of stage-structured metapopulation dynamics.

II. The open-source GeneDrive.jl package furnishes a three-part framework for the exploration and analysis of scenarios wherein organisms are subjected to anthropogenic and environmental change. Its components include (i) a data model to store information unique to genetic and ecological details, (ii) a dynamic model comprised of ordinary differential equations, and (iii) a decision model that discretizes the system of ODEs to formulate a nonlinear mathematical program. Written in the Julia programming language, GeneDrive.jl enables replicable, scalable, and extensible computational experiments by drawing on several state-of-the-art tools within the Julia ecosystem, including its package manager, capacity for multiple dispatch, and the robust suite of numerical solution methods available in OrdinaryDiffEq.jl and JuMP.jl respectively.

III. A principal novelty of GeneDrive.jl is the facility and conceptual clarity it offers in moving scientific experimentation from the wet lab to the dry lab to the field: the data model permits a one-time specification of empirical data on which both ODE and optimization solving algorithms can be called, allowing for the iterative testing of constraints and objectives in the decision model followed by the deployment of optimal policies in the dynamic model. GeneDrive.jl is the first biocontrol-relevant software to feature such a “testbed” functionality. It is also the first with the capacity for optimization, opening this domain area to the opportunities afforded by operations research methods.

IV. An example workflow demonstrates data model, decision model, and dynamic model creation and application. It shows the replacement of vector populations with genetically modified mosquitos that cannot carry disease, followed by an exhibition of the adjustments necessary when the same intervention is conducted under environmental perturbation. This workflow showcases how the inherent composability of GeneDrive.jl can be deployed for relevance to a range of ecological applications.

1. Introduction

Ecology is a complex science, and the field data important to informing its theory is notorious for both sparsity and noise. As a result, concepts fundamental to the population dynamics of many species remain poorly understood. Meanwhile, existing uncertainties are being amplified by the accelerating rate of environmental change and its largely unknown, compounding effects on biological systems (Nowakowski et al. 2018), (Higgins et al. 2021). The scientific challenge extends beyond simply discerning such shifting interactions, however, to actively managing them using biocontrol interventions (Sun et al. 2022), (Monticelli et al. 2022). The spatial and organizational scale of ecological problems lend themselves to computational modelling; such *in silico* experimentation has become an increasingly important and viable technique with which to probe our surrounding world as numerical methods and simulation tools improve (Pascual 2005). This, together with the progressive democratization of computational resources – both insofar as skills-based programming education in the biological sciences and access to computers – has seeded a wealth of ecologically-relevant model development.

GeneDrive.jl employs a stage-structured set of equations to model the metapopulation dynamics of metamorphosing species. This open-source software library – written in the Julia programming language and exploiting many aspects of that rich ecosystem – features a three-part framework (**Figure 1**) with functional applications unique to any existing package (Vasquez 2022). Distinguishing attributes include its robust data model, an abstraction that uses the Julia type system to store simulation inputs and dispatch methods, and the explicit separation of simulation and solving algorithm. The latter underpins a key aspect of GeneDrive.jl: once constructed using the data model, problem specifications can be solved with a variety of either ordinary differential equation methods in the dynamic model or optimization methods in the decision model.

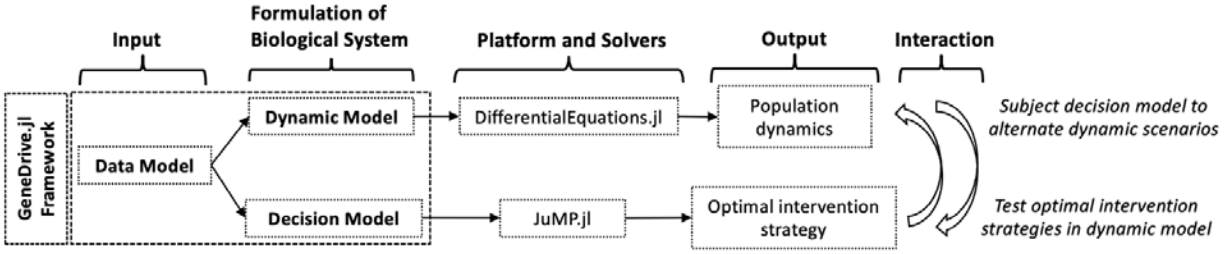


Figure 1: *Overview of the package. GeneDrive.jl is a three-part framework (Data Model, Dynamic Model , and Decision Model).*

This modular architecture – which in the spirit of scientific computing principles is reproducible[1], extensible[2], and scalable[3] – facilitates the empirical pipeline from the wet lab to the dry lab to the field and offers conceptual clarity in doing so: data from the wet lab is used to populate the GeneDrive.jl data model. The data model then parameterizes dry lab experimentation in the dynamic and decision models. The output of these offer guidance for field trials, from which information in turn can be gathered to update values in the data model. Beyond traditional research applications for examining ecological complexities, this workflow can assist with actionable intervention recommendations.

[1] Reproducibility: Given the same inputs, scientists using the same computational tools should be able to achieve the same outputs.

[2] Extensibility: Scientific software should be adaptable to include future, as-yet undefined applications. This is an especially important aspect as the science of genetic biocontrol, including gene drive, continues to accelerate.

[3] Scalability: The systems being examined can be enlarged or their organizational complexity increased to study, e.g., more organisms or organismal interactions over larger areas.

The underlying mathematics of the dynamic and decision models, which are formulated using the same system of population equations, enable a novel "testbed" functionality in GeneDrive.jl that encourages experimentation with operational levers to achieve population management goals in addition to biological ones: it allows for the iterative testing of constraints and objectives in the decision model followed by the deployment of optimal policies in the dynamic model. GeneDrive.jl is the first biologically relevant software to furnish optimization as an option, exposing this domain area to the scientific investigations afforded by operations research methods. Sample parameters and empirically derived functional forms are directly included, with citations, in the GeneDrive.jl data files to provide users with initial values for experimentation. Example information includes genetic, biological, and climatological data.

While GeneDrive.jl is applicable to exploring a range of scenarios, including the spatiotemporal propagation of genetic material and organismal dynamics subjected to anthropogenic and environmental change, it is named in honor of a new frontier in intervention technology. Gene drives – DNA sequences engineered to spread at higher frequencies than Mendelian inheritance patterns – have introduced a revolutionary era in the field of genome engineering (Doudna and Charpentier 2014). The precise editing enabled by CRISPR-Cas9[4] has led to the development of nonlocalized drives as well as drive varieties designed for limited lifespans or localized geographic reach. This active area of research is producing new options for both suppression and replacement-based biological control, promising means for augmenting intervention methods that are now falling short in domains as diverse as conservation, agriculture, and public health. Computational models are key to the study of gene drive organisms and other biocontrol mechanisms as cost-effective and safe precursors to semi or full-scale field trials. In turn, the principles of scientific computing are essential to ensure that the explorations conducted using them can be verified and updated with the rapidly evolving state of knowledge.

2. Materials and methods

Data Model

GeneDrive.jl is predicated on composability. This system design principle is grounded in its data model: an abstraction developed using the Julia type system that standardizes the organization of information, encodes relationships, and permits the assignment of methods (Hay 1996). By enforcing consistency in the specification of data across computational experiments, the data model facilitates reproducibility and data sharing (Sandve et al. 2013). The modularity it enables promotes code re-use as different research questions or focal areas are studied: required inputs may be swapped out accordingly (Gentleman et al. 2004).

The data model includes details that are both exogenous and endogenous to metapopulations. These features, which may be assembled and augmented in a "building block" fashion to develop many unique scientific explorations, are encapsulated in thematic components: climatological, organismal, geographic, and anthropogenic. The panels of **Figure 2** visualize the nested nature of these data model components and show annotated examples of structs from the GeneDrive.jl application programming interface (API), respectively.

[4] CRISPR-Cas9: Clustered regularly interspaced short palindromic repeats-CRISPR-associated 9

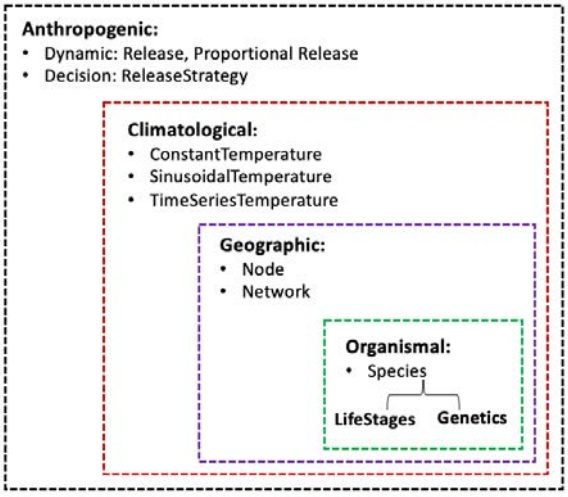


Figure 2A: *The structure of the GeneDrive.jl data model.*

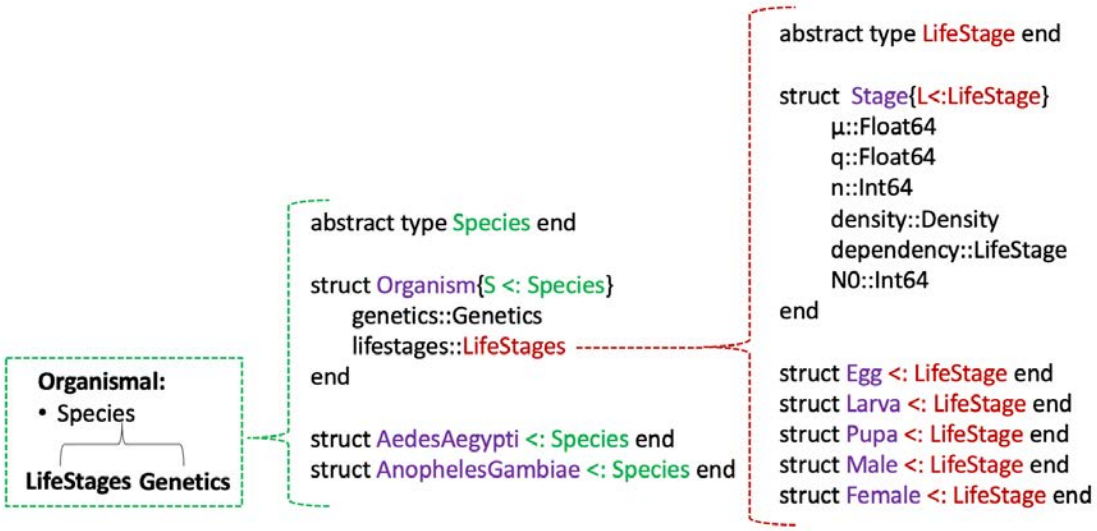


Figure 2B: *Examples of nested structs from the GeneDrive.jl API.*

Experimentation with alternative environmental assumptions can be conducted by parameterizing one of the three `Temperature` structs. These include `ConstantTemperature` to specify a static thermal environment, `SinusoidalTemperature` to furnish an idealized, seasonally fluctuating regime, and `TimeSeriesTemperature` in which vectors of daily values can be stored. Each of these descriptions of thermal trend can be accentuated with heatwaves and cold snaps using the `TemperatureShockData` struct to impose time-bound increases or decreases in temperature.

On the organismal front, the `Genetics` data type defines the likelihood with which offspring will be produced. It also includes information unique to specified genotypes including fertility rates, sex ratios, and varied fitness costs (e.g., the degree to which fecundity for modified organisms is biased with respect to their wild counterparts). The `Organism` type in GeneDrive.jl includes this modular genetic component along with a `LifeStages` data container for details such as stage-specific development time and mortality rate. For select species, empirically derived functions rather than static parameter values are furnished to characterize vital rate responses to environmental perturbations.

Study species defined by the GeneDrive.jl data model inhabit a `Node` - a single homogenous habitat characterized by `Organisms`, `Temperature`, geographic coordinates, and location name - or a heterogeneous `Network`. The `Network` is a collection of unique interconnected `Node`s where movement between each `Node` is specified according to species, life stage, and genotype. This feature enables users to account for a diversity of demographic and genetic migration tendencies as well as exogenous factors that may transport organisms from location to location.

The dynamics of anthropogenic actions such as biological control may be studied in GeneDrive.jl by adding organisms at periodic intervals using the `Release` object. When specifying an intervention schedule, fixed or variably-sized releases can occur at flexible intervals. Alternatively, interventions may be conducted in an adaptive manner that directly accounts for the size of the standing wildtype population using the `ProportionalRelease` approach. Here, the release size is defined according to its relative magnitude with respect to the population of interest at a given timestep (e.g., 20% of wild females present on day 300 of the simulation). As discussed in subsequent sections of this paper, experiments concerning the optimization of anthropogenic interventions also interface with the data model: the `ReleaseStrategy` struct stores information about operational requirements and limitations for use by the decision model.

Exploiting Julia's type system for the GeneDrive.jl data model enables the use of multiple dispatch to assign methods based on function arguments. This powerful feature of the underlying programming language simplifies the flexible expression of characteristics within the population being modelled because, for example, each species or genotype within a single system of interest may be defined by unique responses to the same stimuli (Drake 2005). Alternatively, ambient environmental conditions may be defined and confined to specific nodal habitats within the larger networked metapopulation being studied. Thus, multiple dispatch enables myriad ecological explorations.

Dynamic Model

The dynamic model in the GeneDrive.jl framework is a mathematical representation of continuous-time, stage-structured population dynamics using ODEs. This formulation builds on the widely used lumped age class technique developed by Nisbet and Gurney, where all organisms within a particular life stage are assumed to be equal with respect to their birth, death, and maturation rates (Gurney, Nisbet, and Lawton 1983). Following the multiple examples in ecology and public health applications where Erlang distributions are used to incorporate flexible dwell times, the delay differential equations (DDEs) of Nisbet and Gurney are approximated as a system of ODEs, allowing GeneDrive.jl dynamics to retain biological correctness while sidestepping the modeling complications of DDEs (Hale and Verduyn Lune 1993; Hancock and Godfray 2007; Wu et al. 2021). This facilitates experimentation with the juvenile life stages of ectothermic organisms whose durations are environmentally influenced (Hurtado and Kirosingh 2019; Schneider and Ferris 1986). The initialization routine, common to both the dynamic and decision models, solves for the stable equilibrium of the system given an initial value for the adult female population. **Equation Block 1** specifies the population dynamics of GeneDrive.jl.

$$\omega_g = \sum_{i=1}^{N} \beta_g \sigma_g \left(\mathbf{\Gamma}_g \odot \mathbf{T}_g\right)_i F_i \qquad \forall g \qquad (1a)$$

$$\frac{dE_{g,1}}{dt} = \omega_g - E_{g,1}(\mu_E + q_E n_E) \qquad \forall g \qquad (1b)$$

$$\frac{dE_{g,i}}{dt} = E_{g,i-1} q_E n_E - E_{g,i}(\mu_E + q_E n_E) \qquad \forall g, i = 2 \dots n_E \qquad (1c)$$

$$\frac{dL_{g,1}}{dt} = E_{g,n_E} q_E n_E - L_{g,1}(\mu_L d + q_L n_L) \qquad \forall g \qquad (1d)$$

$$\frac{dL_{g,i}}{dt} = L_{g,i-1} q_L n_L - L_{g,i}(\mu_L d + q_L n_L) \qquad \forall g, i = 2 \dots n_L \qquad (1e)$$

$$\frac{dP_{g,1}}{dt} = L_{g,n_L} q_L n_L - P_{g,1}(\mu_P + q_P n_P) \qquad \forall g \qquad (1f)$$

$$\frac{dP_{g,i}}{dt} = P_{g,i-1} q_P n_P - P_{g,i}(\mu_P + q_P n_P) \qquad \forall g, i = 2 \dots n_P \qquad (1g)$$

$$\frac{dm_g}{dt} = P_{g,n_P} q_P n_P (1 - \theta_g) - m_g \mu_m \qquad \forall g \qquad (1h)$$

$$X_g = P_{g,n_P} q_P n_P \theta_g \frac{m_g \eta_g}{\sum_{k=1}^{N} m_k \eta_k} \qquad \forall g \qquad (1i)$$

$$\frac{dF_{g,i}}{dt} = X_{g,i} - F_{g,i} \mu_F \qquad \forall g, i \qquad (1j)$$

Equation Block 1: ODE system of equations for a single-node formulation of the stage-structured population model. Egg, larval, and pupal stages are represented by E, L, and P respectively, and adults are evenly divided between males, M and females F.

The variable ω_g represents the number of eggs laid, β_g and σ_g are female and male fecundity parameters respectively, and $\mathbf{\Gamma_g}$ and $\mathbf{T_g}$ convey the inheritance and survival probability of the specified genotypes. The formulation of Equations 1(a) and 1(i) follow Sánchez et al (2019).

Mortality rates μ and development rates q are dynamically calculated according to environmental inputs. Logistic density dependence d is implemented in the larval stage L.

The equations of the dynamic model are applicable across metamorphosing species[5]; organism characteristics are specified by the values populating the data model. The approach to parameterization is an important aspect of the empirically informed logic governing GeneDrive.jl's design: the data model establishes an experimental record of the parameters used to initialize a simulation. However, these values may be updated over the course of continuous real-time simulation, facilitating experiments that explore the effect of exogenous perturbations to the biological system being modelled and opening the possibility for new scientific questions to be asked. One example use case is introducing a new organism type into the simulation after it has been initialized; another is abruptly altering the environmental conditions to which simulated organisms are responding. Such options may be useful for investigating, respectively, species invasions relevant to biodiversity explorations or unexpected temperature

[5] Metamorphosis is a developmental process undergone by some species. Holometabolous is known as "complete" metamorphosis; it includes four life stages. Hemimetabolous is also called "simple" or "gradual" metamorphosis; it includes three life stages.

variability broadly relevant to ecological dynamics under climate change. This functionality stems from using discrete callbacks to the solver as explained by the information flow diagram in **Figure 3**, a functionality of the DifferentialEquations.jl platform upon which GeneDrive.jl is constructed and another unique characteristic of the framework presented here.

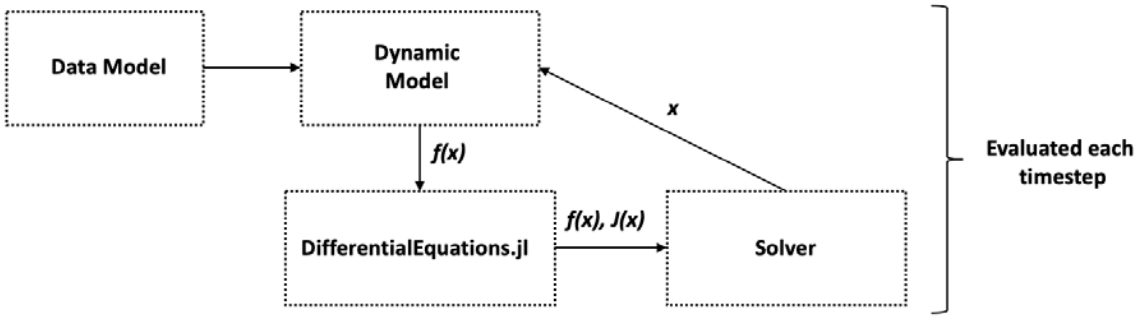


Figure 3: *Dynamic model information flow*

When population dynamics occur across a network, immigration and emigration between nodes in GeneDrive.jl are specified using a transition matrix developed according to Continuous Time Markov Chain (CTMC) theory (Resnick 1992). Each species, life stage, genotype, and adult sex is assigned a rate of dispersal that defaults to zero and may be updated by the user to instantiate spatial interconnections via migration. Whether driven by natural dispersal or the human-induced movement called batch migration, movement rates are generally important to exploring ecological and demographic questions. When the population of interest are modified organisms, the ability to granularly experiment with migration is an essential functionality for analyzing potential risks: it can help scientists understand the role that landscape changes, species adaptation, or genetic evolution might play in the spatial reach of transgenes (Tanaka, Stone, and Nelson 2017; Huestis et al. 2019).

Composability in GeneDrive.jl extends to the solution methods available to both simulation arms of the framework. Because the dynamic model is built on the DifferentialEquations.jl platform, users may choose algorithms from the robust suite of options compatible with that package or included in its common interface; Tsit5, an explicit Runge-Kutta method, is recommended for most non-stiff systems such as that formulated in GeneDrive.jl. In general, ODE solver selection is guided by a user's desire for adaptive versus non-adaptive time-stepping of the integrator, accuracy as dictated by relative and absolute tolerances – the allowed error in a single step of the integrator – and the stiffness of the system.

Decision Model

The decision model in the GeneDrive.jl framework is formulated as a nonlinear program (NLP). Mathematical programming is a method that determines the optimal set of control (also called decision) variable values required to meet a given objective (goal). The objective is generally specified using a linear or quadratic function; this function is then either maximized or minimized. The "best" set of values to satisfy an objective function are often referred to as the optimal schedule or optimal policy and are subject to constraints: limitations that restrict the feasible solution space. Constraints are defined using mathematical equalities or inequalities.

The GeneDrive.jl decision model is implemented by discretizing the dynamic model's system of ODEs using the Euler approximation method and a daily timestep, which was selected to reflect the one-day

time constant common to empirically collected data values in ecology. The discretized population equations thus constitute the equality constraints of the resulting optimization problem and inform the feasibility of a given simulation with experiment-specific biological details. Like the dynamic model, decision model parameters are populated by data model values and equality constraints are applicable to alternative hemimetabolous or holometabolous species according to this parameterization. Unlike the dynamic model, the information in this simulation is evaluated once and over the full time horizon, as depicted in **Figure 4**.

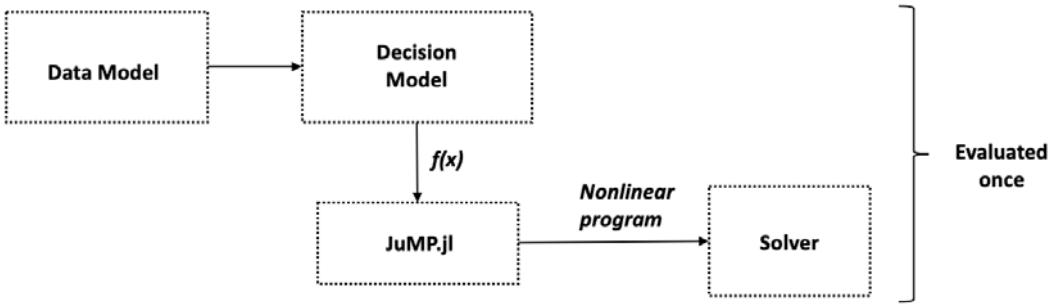


Figure 4: *Decision model information flow*

Because the equality constraints are comprised of the discretized population equations of the ODE model, in the absence of an objective function, the population dynamics output by the decision model are qualitatively comparable to those of the dynamic model. This enables iterative experimentation wherein, e.g., optimal intervention policies output by the decision model may be tested under alternative conditions in the dynamic model. This utility is demonstrated in the "Results" section. When the problem of interest is defined over a network, the transition matrix is again used to describe species and genotype-specific rates.

Constraints representing non-biological or operational limitations, such as resource availability and geographic reach, enter the GeneDrive.jl decision model as inequality constraints. These are defined by the user in the `ReleaseStrategy` struct of the data model, so named in homage to the biocontrol intervention strategies commonly used in the public health, agriculture, and invasive species arenas wherein organisms modified for e.g. sterility are released into the environment to reduce the standing wildtype population of a given species (Heimpel and Mills 2017). The values of this struct are assigned on a per-node basis in the case of a network implementation, enabling spatially explicit constraints and the exploration of myriad combinations of policies. Default values are supplied where no information is specified; this information is accessible by viewing the fields of the data model.

The default nonlinear solver in GeneDrive.jl is the powerful and free Ipopt. This choice may be updated by users interested in exploring different solvers or augmenting the Ipopt optimizer with additional algorithms to accelerate solutions or improve accuracy. Ipopt relies on third party code for obtaining the solution of "sparse, symmetric, indefinite linear systems"(Waechter and Laird 2022); to this end, various alternative free and commercial linear solvers can be chosen from among a list in the online documentation to speed results.

Optimization as commonly implemented in the biological sciences employs continuous time models and thus requires using the Hamiltonian[6] to find the adjoint equation[7] for each state considered[8] (Khamis et al. 2018; Agusto and Khan 2018). However, this conventional approach may require complex reformulations; the derivation of the Hamiltonian can quickly become intractable (Lenhart and Workman 2007). Applying such methods to ecological systems, which may for example include several organismal life stages and many possible genetic variations of those stages that vastly expand the state space, demands significant simplification and the omission of details that are potentially critical to scientifically informative results.

By comparison, the degree of mathematical sophistication required for numerically solving continuous models as discretized NLPs is notably lower. This more accessible method also accommodates the inclusion of a larger state space and thus significantly more detail, allowing for more realistic models and consequently more informative results. The solutions returned by available software libraries for the largescale optimization of continuous systems such as Interior Point OPTimizer (IPOPT) furnish the guarantee of finding an optimum, and in the case of infeasibility state as much. In addition, objective functions and constraints are straightforward to implement and test, accelerating model iteration to the advantage of empirical exploration.

3. Results

Example GeneDrive.jl workflow

A GeneDrive.jl workflow is demonstrated here, from data model creation to simulation in the dynamic and decision models respectively. In the public health arena, a common motivation for biological control is to reduce or eliminate potential disease vectors. Female mosquitoes of the *Aedes* genus represent one of the most prevalent human health threats in the world; the blood meals they require to reproduce can spread deadly illnesses such as dengue and Zika (Powell 2018). One genetic-based technology already in use to reduce this risk in locations like Brazil and the United States – and which, like some gene drive options, employs a suppression approach – is called Release of Insects with Dominant Lethal (RIDL) (Carvalho et al. 2015). The subsequent problem demonstrates how releasing RIDL-modified male mosquitoes in large and regular quantities can lower (suppress) the population level of adult female vectors.

The example shown first defines a data model using the vital rates, thermal biology, genetic characteristics, and hypothetical habitat of an *Aedes aegypti* mosquito. The demonstrated workflow draws from pre-constructed data models, highlighting the code brevity and simplicity made possible in the GeneDrive.jl design – particularly once problem information has been assembled and stored. All data used in this case study are stored in the GeneDrive.jl package for replication and extension.

[6] The Hamiltonian is a function used to solve a problem of optimal control for a dynamical system; it is an instantaneous increment of the Lagrange expression of the problem to be optimized over the time horizon (Ferguson and Lim 1998).

[7] The adjoint can be interpreted as Lagrange multipliers associated with the state equations, where the states represent constraints of the minimization problem, and the adjoints represent the marginal cost of violating those constraints (Takayama 1985).

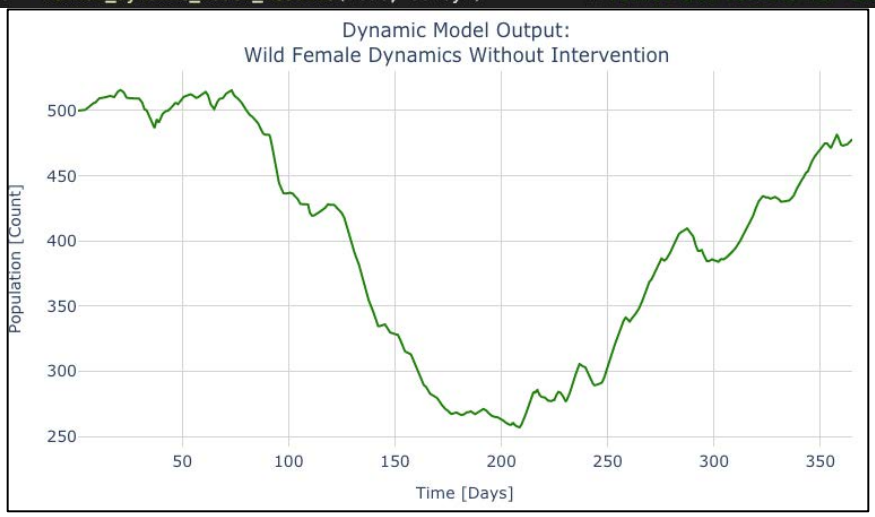
[8] The technique, as applied to deterministic ordinary differential equations in the context of minimization, is to solve a set of necessary conditions to be satisfied by the optimal control u^* and corresponding state x^*. Those necessary conditions are derived according to the inverse of the inequality in the Pontryagin Maximum Principle. This principle converts the problem into one of minimizing the Hamiltonian pointwise with respect to the controls (Ross 2009).

```
species = AedesAegypti                                          # Select species type
genetics = genetics_ridl();                                     # Define how genetic information is passed on
enviro_response = stages_rossi();                               # Choose functional form of environmental response
update_population_size(enviro_response, 500);                   # Update population size
organisms = make_organisms(species, genetics, enviro_response); # Assemble organism
temperature = example_temperature_timeseries;                   # Define temperature data
coordinates = (16.1820, 145.7210);                              # Specify the geographic coordinates
node = Node(:YorkeysKnob, organisms, temperature, coordinates); # Define geographic type, name location, "populate" it

op_constraints = ReleaseStrategy(release_this_gene_index = 3,   # Include operational constraints
    release_this_life_stage = Male,
    release_time_interval = 7,
    release_start_time = 10,
    release_size_max_per_timestep = 50000.0)
mystrategy = Dict(1 => op_constraints);
```

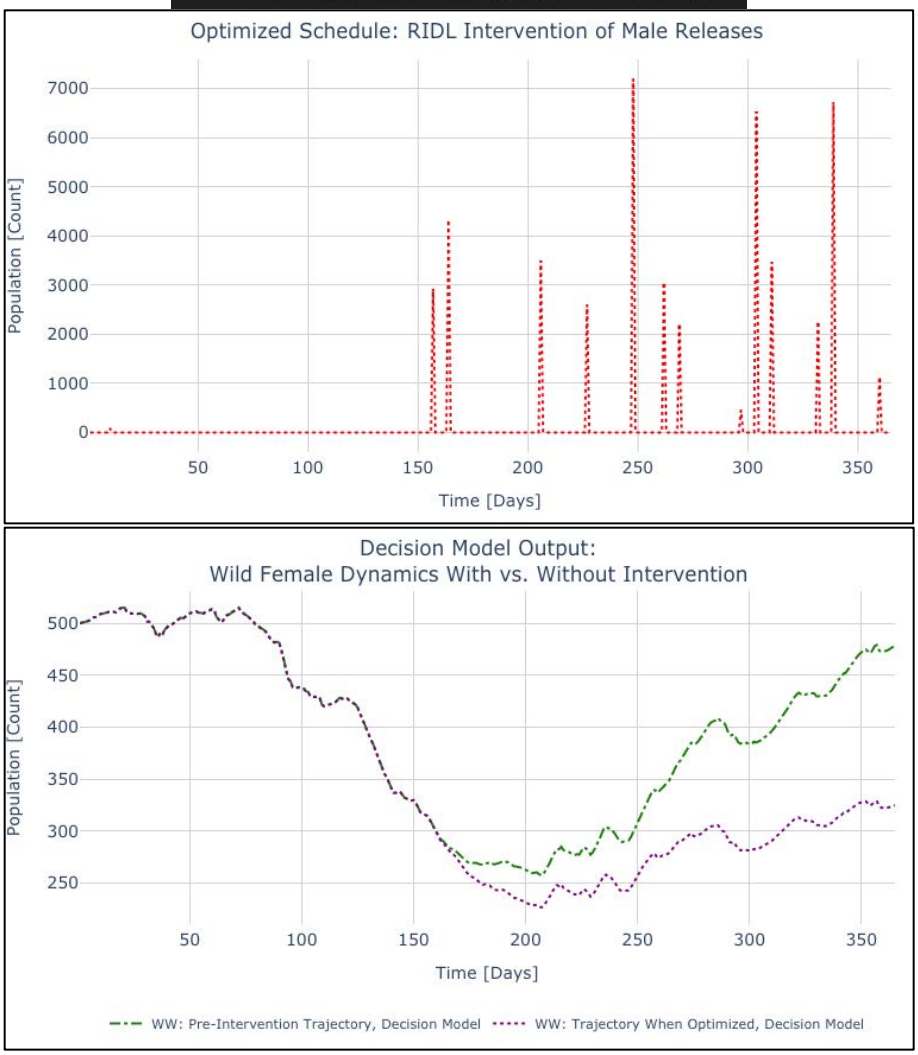
Next, this data model is used to parameterize a year-long simulation in the dynamic model. In visualizing a subset of the results, we see that the wildtype adult female population fluctuates in response to the daily and seasonal shifts in temperature.

```
tspan = (1,365)                                                 # Define the problem time horizon
solver = OrdinaryDiffEq.Tsit5()                                 # Select solver
soldyn = solve_dynamic_model(node, solver, tspan);              # Solve
results = format_dynamic_model_results(node, soldyn)            # Format all results for analysis
```



Here, an objective function is selected. The objective function, which is subject to the biological constraints represented by a discretized iteration of the equations in **Equation Block 1**, is also subject to the operational constraints previously specified in the data model using the `ReleaseStrategy` struct. Both objective function and operational constraints are input as arguments to the decision model, which itself is comprised of the biological constraints. An optimal schedule (timing and size) for releasing RIDL-modified organisms is produced; this schedule is visualized in red below. The population dynamics output by the decision model upon applying this schedule are visualized with a purple dotted line; the original population dynamics of the decision model pre-optimization are visualized with the green dashed line. The operational constraints and objective function are expressed mathematically in **Equation Block 2**.

```
soldec = solve_decision_model(prob,
    TargetPercentageByDate; wildtype= 1,
    percent_suppression=.40, target_timestep=200)
results = format_decision_model_results(soldec)
```

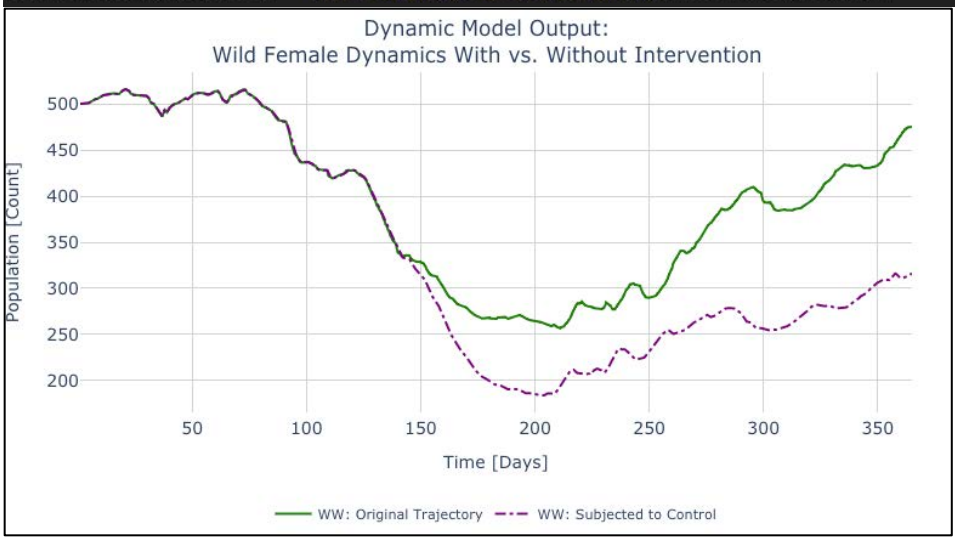


$$\min_{c_{g^*,t}} \quad \sum_{t=1}^{N} \left((F_{\hat{g},t} - \gamma F_{\hat{g},1})^2 + \alpha c_{g^*,t} \right) \qquad (2)$$
$$\text{s.t.} \quad c_{g^*,t} \leq R_{max}^{day} \quad \forall t \in T_r$$

Equation Block 2: Objective function and operational constraints. Where R_{max}^{day} is the maximum allowed releases per day, $\mathcal{T}_r \subset \mathcal{T}$ is defined as the days when releases are allowed, and the modified genotype for release is c_{g^*}.

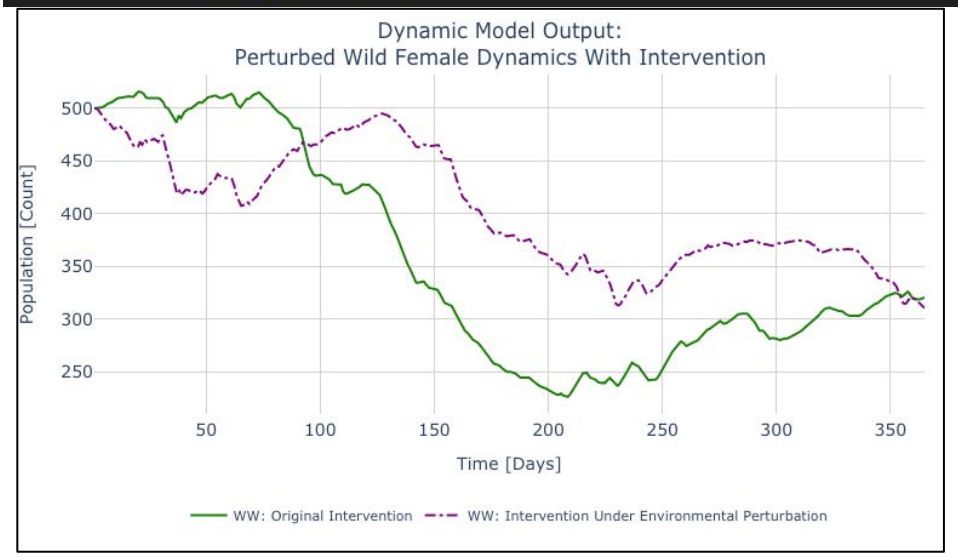
In the following code block and visualization, the solution of the optimization problem is explored using the ODE model: the optimal policy of release size and timing produced by the decision model is input to the dynamic model using the `Release` struct. A simple timeseries plot demonstrates that the resulting dynamics between the two simulation arms of GeneDrive.jl are indeed comparable both pre- and post-intervention (see, respectively, the green solid line and purple dashed line resulting from the dynamic model below compared to the corresponding colors in the output of the decision model above).

```
releases = Release(node, species, Male, 3, Float64.(times), Int64.(values));
soldyn_opt1 = solve_dynamic_model(node, [releases], solver, tspan);
results_optimalsched1 = format_dynamic_model_results(node, soldyn_opt1)
```



To experiment with the effect of the optimal solution being deployed under different environmental conditions, the original data model is updated by increasing the daily ambient temperature of the *Aedes aegypti* habitat by 2°C. When the optimized intervention schedule is re-applied featuring this new parameterization of the dynamic model, the resulting behavior (purple dashed line) in the visualization below differs markedly from the earlier output (green solid line): using the same releases, the disease vector population under the new temperature regime does not respond equivalently.

```
hotter_temperature = perturb_temperature_timeseries(temperature, 2.0);
node_PERT = Node(:YorkeysKnob, organisms, hotter_temperature, coordinates);
soldyn_PERT = solve_dynamic_model(node_PERT, [releases], solver, tspan);
```

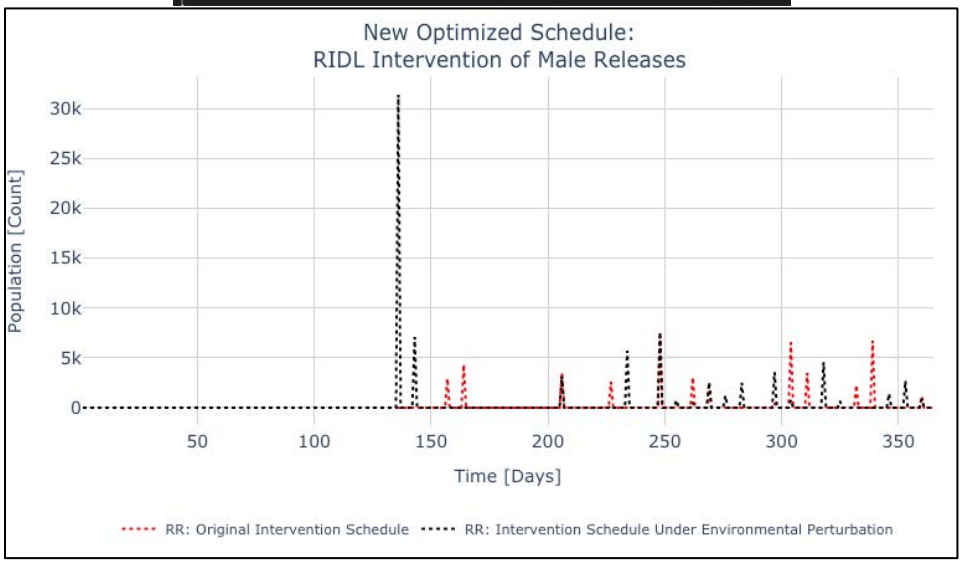


However, re-running the optimization model using the altered temperature regime parametrization produces an updated policy of optimal release sizes and timings; this new schedule can then be tested in the dynamic model to ensure that population fluctuations there match those produced by adult females in the decision model. Below, we see the new optimal set of decision variables compared to those

generated under the original environmental regime and note the distinctions, both in timing and size of releases, that are demanded by the two different temperature scenarios.

```
prob_PERT = create_decision_model(node_PERT, tspan;
    node_strategy = mystrategy,
    species = AedesAegypti,
    optimizer = i)

soldec2 = solve_decision_model3(prob_PERT,
    TargetPercentageByDate; wildtype= 1,
    percent_suppression=.40, target_timestep=200)
```



4. Discussion

Significance of results

The GeneDrive.jl workflow example employs a case study that illustrates how public health and environmental concerns can overlap. It also demonstrates that there is no such thing as a "one size fits all" approach to intervention planning: biocontrol is a hyper-local undertaking that hinges on regional realities. Simulation models used to understand and manage ecological risks such as the climate-driven expansion of disease vectors must account for this and do so in a way that allows easy iteration as environmental factors change, or as updated safety information and methods for biocontrol become available.

There are basic uncertainties associated with how genetic technologies comparable to RIDL, each designed to exhibit unique characteristics and achieve specific outcomes, might function outside the laboratory. In many cases, these unknowns underscore open questions about the wild organisms that are being modified. Mating behavior, dispersal, density dependence, and overwintering remain poorly understood for many insect species in which genetic tools, including gene drive, are currently being developed. Such questions are additionally complicated by the interactions that might exist between novel genetic material, wildtype organismal populations, and the surrounding environment.

While stochastic simulations are an important means of probing both uncertainty and potential variability, deterministic models such as the current iteration of GeneDrive.jl furnish insights useful to understanding the mean behavior of metapopulations. Sensitivity analyses are one approach to

exploring the range of possible parameter space in deterministic models. The design of GeneDrive.jl enables experimentation with functional form as well as parameterization, addressing the potential concern that mechanistic models may introduce structural bias by mis-specifying dynamics. Under such a scenario, even parameter combinations selected to exemplify extremes may not simulate the full extent of possible outputs. Scientists such as Runge and Johnson suggest that "biological knowledge be used to bracket a range of possible functional forms, and robustness of conclusions [then be] checked over this range" (Runge and Johnson 2002).

The building block-based approach that permits scientists to progressively develop organismal or environmental characteristics and test a variety of formulations derived from field and laboratory data is part of what enhances the empirical nature of GeneDrive.jl. For example, the temperature-responsive functions employed in the case study were calibrated for *Aedes Aegypti* in Brazil (Rossi, Ólivêr, and Massad 2014), but can be exchanged with alternative dynamics – or be made responsive to alternative temperature inputs – by updating a single line in the code in either case. This facet, possible because composability is prioritized across the GeneDrive.jl software stack, renders the featured workflow useful for an array of policy decisions as well as experimental ends.

Related work

Other models also simulate population dynamics and are applicable to studies of biocontrol using gene drive. One, Skeeter Buster, couples a pre-existing and highly detailed model of *Aedes aegypti* weather-driven life cycle dynamics called the Container Inhibiting Mosquito Simulation Model (CIMSiM) with an explicit spatial structure, genetic information, and stochasticity (Magori et al. 2009). However, it is most appropriate for very fine spatial scales such as household-level dynamics. Further, Skeeter Buster is not open source. A second model, EMOD, has been rigorously developed over the course of a decade and includes both spatial and environmental details. It is openly accessible and extensible by users as desired. However, it is an agent-based model primarily designed to track infectious disease transmission between individual humans (Eckhoff 2011). EMOD requires data-rich settings and its "configurability comes at the cost of ease-of-use" (Eckhoff et al. 2017). A third model, SLiM, is extensible to a range of scenarios while also being open source and incorporating both spatial explicitness and environmental variables. But the complexity of its experimental capabilities can render it difficult to use (Haller and Messer 2019).

Fourth and finally, the Mosquito Gene Drive Explorer or MGDrivE1.0 and its recent update, MGDrivE2.0, are designed for the express purpose of studying gene drive (Sánchez C. et al. 2020; Wu et al. 2021). Both are open source and can be user specified; likewise, they explicitly incorporate age structure, genetics, and spatial details. However, these models have been highly optimized to focus on their developers' current research priorities. While the speed of both MGDrivE and MGDrivE2 is driven by an RCPP framework underlying an R wrapper, this configuration complicates users' ability to prototype new experiments. Given the rapidly evolving state of the science in gene drive technology, and the myriad uncertainties associated with using it as well as alternative means of biocontrol in realistic ecological settings, there is a clear need for fast and straightforward iterative capacity such as that brought by GeneDrive.jl. Its dual use design, wherein problems once defined may be either simulated or subject to optimization, enables experimental setups that are not possible in any other currently available software for biologically relevant systems.

5. Conclusion

GeneDrive.jl permits the study of metapopulation dynamics, allowing users to examine how these dynamics contribute to the invasion, persistence, or extinction of genotypes. This is a functionality applicable to many areas of biological science but particularly lends itself to the study of new genetic tools for biocontrol such as gene drive. The software empowers computational experiments that reflect the traditional scientific process – data gathering, hypothesis testing via simulation, and an iterative analysis of results to support or discard that hypothesis. In the realm of biological control, such a workflow is essential to ensuring that both the benefits and risks of interventions are considered as research moves from the laboratory to field testing and eventually to widespread application. The public sensitivities surrounding gene drive technology in particular demand models that are more than just mathematically consistent and computationally efficient: they must be straightforward to understand and use, to allow continued extensions and improvements by multidisciplinary research communities as new insights are gained.

Beyond the philosophy of design that guided its development, the Julia programming language ingrains scientific computing principles into the GeneDrive.jl framework. On the reproducibility front, Julia's package manager supplies the up-to-date versions of all relevant dependencies and records compatibility constraints (JuliaLang 2022). This vastly eases user experience by rendering a "ready-made" working environment. Julia's ecosystem also allows the use of several analysis tools historically familiar to the biologists, ecologists, and computational scientists that are the intended end-users of GeneDrive.jl, including full libraries for R, Python, Java, MATLAB, Mathematica, and C. A large suite of fast Julia-wrapped and native solvers enforces the separation between the simulation of interest and the solution method, allowing the user to focus exclusively on scientific details without resorting to the development of their own solving algorithms or paying for expensive commercial solvers unless desired. Finally, Julia boasts a quickly growing community of developers interested in biological questions and expressly focused on cutting edge computational advances in differential equation solution methods and mathematical programming, among other areas (Rackauckas and Nie 2017; Dunning, Huchette, and Lubin 2017; Bezanson et al. 2017).

By building on free and open source platforms, offering the option for fast prototyping that can be easily scaled, and facilitating the integration of empirically derived functional forms that reflect the latest biological understanding (e.g., of thermal biology), GeneDrive.jl promotes the use of mathematical models for evidence-based decision making. Such tools are important for advances in the field as well as in the policy realm; it is essential that they remain inclusive of users in low income settings and accessible to those with comparatively little programming experience. GeneDrive.jl also renders a powerful method for optimization, nonlinear programming, immediately available for the design of interventions. Future work will incorporate stochasticity to investigate questions of risk, expand test coverage, and further modularize the population equations underlying the dynamic and decision models such that non-metamorphosing species may be simulated.

References

Agusto, F. B., and M. A. Khan. 2018. "Optimal Control Strategies for Dengue Transmission in Pakistan." *Mathematical Biosciences* 305: 102–21.

Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B. Shah. 2017. "Julia: A Fresh Approach to Numerical Computing." *SIAM Review* 59 (1): 65–98.

Carvalho, Danilo O., Andrew R. McKemey, Luiza Garziera, Renaud Lacroix, Christl A. Donnelly, Luke Alphey, Aldo Malavasi, and Margareth L. Capurro. 2015. "Suppression of a Field Population of Aedes Aegypti in Brazil by Sustained Release of Transgenic Male Mosquitoes." *PLoS Neglected Tropical Diseases* 9 (7): e0003864.

Doudna, Jennifer A., and Emmanuelle Charpentier. 2014. "The New Frontier of Genome Engineering with CRISPR-Cas9." *Science* 346 (6213): 1258096.

Drake, John M. 2005. "Population Effects of Increased Climate Variation." *Proceedings of the Royal Society B: Biological Sciences* 272 (1574): 1823–27.

Dunning, Iain, Joey Huchette, and Miles Lubin. 2017. "JuMP: A Modeling Language for Mathematical Optimization." *SIAM Review* 59 (2): 295–320.

Eckhoff, Philip A. 2011. "A Malaria Transmission-Directed Model of Mosquito Life Cycle and Ecology." *Malaria Journal* 10 (1): 1–17.

Eckhoff, Philip A., Edward A. Wenger, H. Charles J. Godfray, and Austin Burt. 2017. "Impact of Mosquito Gene Drive on Malaria Elimination in a Computational Model with Explicit Spatial and Temporal Dynamics." *Proceedings of the National Academy of Sciences* 114 (2): E255–64.

Ferguson, B.S., and G. Lim. 1998. *Introduction to Dynamic Economic Problems*. Manchester: Manchester University Press.

Gentleman, Robert C., Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, et al. 2004. "Bioconductor: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology* 5 (10): R80. https://doi.org/10.1186/gb-2004-5-10-r80.

Gurney, W. S. C., R. M. Nisbet, and J. H. Lawton. 1983. "The Systematic Formulation of Tractable Single-Species Population Models Incorporating Age Structure." *The Journal of Animal Ecology*, 479–95.

Hale, Jack, and Jjoerd Verduyn Lune. 1993. *Introduction to Functional Differential Equations*. 1st ed. Applied Mathematical Sciences 0066–5452.

Haller, Benjamin C, and Philipp W Messer. 2019. "Evolutionary Modeling in SLiM 3 for Beginners." *Molecular Biology and Evolution* 36 (5): 1101–9. https://doi.org/10.1093/molbev/msy237.

Hancock, Penny A., and H. Charles J. Godfray. 2007. "Application of the Lumped Age-Class Technique to Studying the Dynamics of Malaria-Mosquito-Human Interactions." *Malaria Journal* 6 (1): 1–13.

Hay, David C. 1996. *Data Model Patterns: Conventions of Thought*. Dorset House Pub.

Heimpel, George E., and Nicholas J. Mills. 2017. *Biological Control*. Cambridge University Press.

Higgins, Emma A., Doreen S. Boyd, Tom W. Brown, Sarah C. Owen, and Adam C. Algar. 2021. "Disentangling Controls on Animal Abundance: Prey Availability, Thermal Habitat, and Microhabitat Structure." *Ecology and Evolution* 11 (16): 11414–24. https://doi.org/10.1002/ece3.7930.

Huestis, Diana L., Adama Dao, Moussa Diallo, Zana L. Sanogo, Djibril Samake, Alpha S. Yaro, Yossi Ousman, Yvonne-Marie Linton, Asha Krishna, and Laura Veru. 2019. "Windborne Long-Distance Migration of Malaria Mosquitoes in the Sahel." *Nature* 574 (7778): 404–8.

Hurtado, Paul J., and Adam S. Kirosingh. 2019. "Generalizations of the 'Linear Chain Trick': Incorporating More Flexible Dwell Time Distributions into Mean Field ODE Models." *Journal of Mathematical Biology* 79 (5): 1831–83.

JuliaLang. 2022. "Pkg · The Julia Language." May 11, 2022. https://docs.julialang.org/en/v1/stdlib/Pkg/#Pkg-1.

Khamis, Doran, Claire El Mouden, Klodeta Kura, and Michael B. Bonsall. 2018. "Optimal Control of Malaria: Combining Vector Interventions and Drug Therapies." *Malaria Journal* 17 (1): 1–18.

Lenhart, Suzanne, and John T. Workman. 2007. *Optimal Control Applied to Biological Models*. Chapman and Hall/CRC.

Magori, Krisztian, Mathieu Legros, Molly E. Puente, Dana A. Focks, Thomas W. Scott, Alun L. Lloyd, and Fred Gould. 2009. "Skeeter Buster: A Stochastic, Spatially Explicit Modeling Tool for Studying

Aedes Aegypti Population Replacement and Population Suppression Strategies." *PLoS Neglected Tropical Diseases* 3 (9): e508.

Monticelli, Lucie S., Jacob Bishop, Nicolas Desneux, Geoff M. Gurr, Coline C. Jaworski, Ailsa HC McLean, Eva Thomine, and Adam J. Vanbergen. 2022. "Multiple Global Change Impacts on Parasitism and Biocontrol Services in Future Agricultural Landscapes." In *Advances in Ecological Research*, 65:245–304. Elsevier.

Nowakowski, A. Justin, Luke O. Frishkoff, Mickey Agha, Brian D. Todd, and Brett R. Scheffers. 2018. "Changing Thermal Landscapes: Merging Climate Science and Landscape Ecology through Thermal Biology." *Current Landscape Ecology Reports* 3 (4): 57–72.

Pascual, Mercedes. 2005. "Computational Ecology: From the Complex to the Simple and Back." *PLoS Computational Biology* 1 (2): e18.

Powell, Jeffrey R. 2018. "Mosquito-Borne Human Viral Diseases: Why Aedes Aegypti?" *The American Journal of Tropical Medicine and Hygiene* 98 (6): 1563–65. https://doi.org/10.4269/ajtmh.17-0866.

Rackauckas, Christopher, and Qing Nie. 2017. "Differentialequations. Jl–a Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia." *Journal of Open Research Software* 5 (1).

Resnick, Sidney I. 1992. *Adventures in Stochastic Processes*. Springer Science & Business Media.

Ross, I. 2009. *A Primer on Pontryagin's Principle in Optimal Control*. Collegiate Publishers.

Rossi, Marcelo Margon, Lêuda Ólivêr, and Eduardo Massad. 2014. "Modelling the Implications of Temperature on the Life Cycle of Aedes Aegypti Mosquitoes." In *Ecological Modelling Applied to Entomology*, 81–107. Springer.

Runge, Michael C., and Fred A. Johnson. 2002. "The Importance of Functional Form in Optimal Control Solutions of Problems in Population Dynamics." *Ecology* 83 (5): 1357–71.

Sánchez C., Héctor M., Sean L. Wu, Jared B. Bennett, and John M. Marshall. 2020. "MGDrivE: A Modular Simulation Framework for the Spread of Gene Drives through Spatially Explicit Mosquito Populations." *Methods in Ecology and Evolution* 11 (2): 229–39. https://doi.org/10.1111/2041-210X.13318.

Sandve, Geir Kjetil, Anton Nekrutenko, James Taylor, and Eivind Hovig. 2013. "Ten Simple Rules for Reproducible Computational Research." *PLOS Computational Biology* 9 (10): e1003285. https://doi.org/10.1371/journal.pcbi.1003285.

Schneider, S. M., and H. Ferris. 1986. "Estimation of Stage-Specific Developmental Times and Survivorship from Stage Frequency Data." *Researches on Population Ecology* 28 (2): 267–80.

Sun, Yan, Tobias Züst, Daniele Silvestro, Matthias Erb, Oliver Bossdorf, Pierre Mateo, Christelle Robert, and Heinz Müller-Schärer. 2022. "Climate Warming Can Reduce Biocontrol Efficacy and Promote Plant Invasion Due to Both Genetic and Transient Metabolomic Changes." *Ecology Letters*.

Takayama, A. 1985. *Mathematical Economics*. Cambridge University Press.

Tanaka, Hidenori, Howard A. Stone, and David R. Nelson. 2017. "Spatial Gene Drives and Pushed Genetic Waves." *Proceedings of the National Academy of Sciences* 114 (32): 8452–57.

Vasquez, Valeri. 2022. *GeneDrive.Jl* (version 0.1.1). Julia. https://github.com/vnvasquez/GeneDrive.jl.

Waechter, Andreas, and Carl Laird. 2022. "Ipopt: Documentation." May 10, 2022. https://coin-or.github.io/Ipopt/.

Wu, Sean L., Jared B. Bennett, Héctor M. Sánchez C, Andrew J. Dolgert, Tomás M. León, and John M. Marshall. 2021. "MGDrivE 2: A Simulation Framework for Gene Drive Systems Incorporating Seasonality and Epidemiological Dynamics." *PLOS Computational Biology* 17 (5): e1009030. https://doi.org/10.1371/journal.pcbi.1009030.