The ODD Protocol

I developed B3GET using the NetLogo modeling environment, which I chose among available microscale modeling software (RePast: Collier, 2003; MASON: Luke et al., 2005; SWARM: Minar et al., 1996) because it includes interface visualization and facilitates educational dissemination. Here, I report on B3GET following the Overview, Design concepts and Details (ODD) description protocol for microscale models (Grimm et al., 2020, 2010, 2006).

1 Purpose and patterns

B3GET is built to simulate virtual biological **`animals`** that respond to ecological challenges in a similar manner to real world animals. These **`animals`** move, eat, reproduce, and perform other behaviors that are key to understanding dynamic biological systems.

2 Entities, state variables, and scales

B3GET includes two types of entities, **`plants`** and **`animals`**, whose state variables are listed in full in Table 1. **`Plants`** are arranged as contiguous cells in the two-dimensional and toroidal environment (100 x 100 cells) that define the landscape on which the **`animals`** exist. **`Plants`** possess a few state variables that by NetLogo's convention begin with the letter "p" for patch. `penergy.supply` defines the current amount of energy that a **`plant`** contains and `pterminal.energy` defines the maximum amount of energy that a **`plant`** can presently achieve by growing. Cells and timesteps represent real-world concepts for space and time but do not map onto specific areas or time units. This allows the user to fully recognize this model as a separate universe with its own set of physical laws, rather than adhering to the laws of the natural world.

`Animals` are generic virtual organisms who survive and reproduce with constraints that approximate the biological constraints of real-world living organisms. **`Animals`** are characterized by several state variables, which are differentiated as (1) attributes that are perceptible to themselves and others, (2) attributes that are only perceptible to themselves, and (3) hidden "tracking" variables that passively accumulate information during their lifetime. The attributes of an **`animal`** form a collection of traits that can be considered the outward expression, or phenotype, of that individual. Perceptible attributes include: `biological.sex` ("`male`" or "`female`"), `body.size`, `body.shade`, `life.history` ("`gestatee,`" "`infant,`" "`juvenile,`" or "`adult`"), whether the individual `is.alive` or `is.resting`, if they have their `yellow.signal`, `red.signal`, or `blue.signal` turned on, and if they have any `carried.items` ("`true`" or "`false`"). All **`animals`** have a `group.identity` state variable, a numerical value that encodes a color to facilitate visually distinguishing **`animals`** by their group membership. A collection of attributes, `visual.angle`, `visual.range`, `day.perception`, `night.perception`, define an **`animal's`** ability to see. **`Animals`** also possess several attributes that only the possessor can detect, including their `female.fertility` ("`cycling,`" "`pregnant,`" or "`lactating`" for an `adult female`) and `bite.capacity`, which determines how much energy can be consumed by a **`plant`** source. **`Animals`** also track the chance of certain events occurring, such as `juvenility.chance` and `weaning.chance`, which influences when an "`infant`" becomes a "`juvenile`" respectively through either an individual or their mother's decision-making. **`Animals`** have an equivalent attribute to **`plants`**, `energy.supply`, in which they store the energy they receive by nursing, eating **`plants`**, or eating each other.

Many of these variables, including `body.size`, `conception.chance`, and `survival.chance`, encode a value between 0 and 1. For interpretation, these values can be rescaled to match any value in nature. These values can be modified throughout an **`animal's`** lifetime depending on how much of their `energy.supply` they or another individual invests in actions to perform this modification. **`Animals`** pay a positive amount of their `energy.supply` to pay for performing behaviors, and are not able to enact these behaviors if their `energy.supply` reaches zero. However, **`animals`** can pass negative values to perform behaviors that produce the opposite effect. For example, paying negative energy toward the *`sex-ratio`* action increases the chance of conceiving a female, whereas paying positive energy toward this action increases the chance of conceiving a male. These modifications apply to the equation below, which can incrementally increase or decrease the original value while remaining within the 0 to 1 range: y_0 is the initial state variable value, x is an `energy.supply` value between -∞ and ∞, and y is the final value.

$$y(x, y_o) = \begin{cases} y_0^{(1+|x|)} & \text{if } x < 0 \\ y_0^{1/(1+x)} & \text{if } x \geq 0 \end{cases}$$

Equation for `Animal` attribute changes. This equation takes as input y_0, a real number between 0 and 1, and x, any real number from negative to positive infinity. Output y is a real number between 0 and 1 such that positive values of x result in $y > y_0$ and negative values of x result in $y < y_0$ and the difference between y and y_0 is proportional to |x|.

Upon conception, **`animals`** have several static variables that remain fixed for their lives, including `biological.sex`, `generation.number`, references to their `mother.identity` and `father.identity`, and `behavior.chromosome` and `identity.chromosome` that contain the list of genes inherited from their parents. **`Animals`** also possess some dynamic variables pertaining to current status, such as `age.in.ticks`, `body.size`, and `life.history` stage, which can change throughout the course of their lives. Furthermore, **`animals`** have spatial coordinates that change during the simulation as they *`move`*. Tracking variables relate to those that do not influence the lives of **`animals`** in any way, but instead track their lives, as if recorded by a passive observer. For example, `ticks.at.weaning` and `ticks.at.sexual.maturity` record the time when an **`animal`** became a juvenile and an adult, respectively.

Table 1 List of `animal` state variables in B3GET

State variable (a)	Description (b)	Value options (c)	Value at conception (d)
`my.identity`	A unique identifying number for all **`animals`**.	Integer	Random integer
(e) Visible attributes			
`biological.sex`	Upon conception, each **`animal`** is assigned a sex based on ratio preference of both parents	"male" or "female"	Probability of either based on parental `sex.ratio` preference.
`life.history`	Upon conception, **`animals`** are assigned "gestatee" status, but are expected to move through each life history stage depending on how much energy is invested to upgrading life history.	"gestatee," "infant," "juvenile," "adult"	"gestatee"

female.fertility	Only adult **females** have a fertility status, which changes depending on events like conception and weaning.	"cycling," "pregnant," or "lactating"	—
group.identity	Each **animal** is affiliated with exactly one group, which can influence spatial movement and interaction decision-making.	Integer	Current group.identity of mother
is.alive	Defines whether or not an **animal** is alive, which allows them to perform actions, and not otherwise.	Boolean	TRUE
yellow.signal	When this variable is set to true, the **animal** has its yellow signal showing on its back.	Boolean	FALSE
red.signal	When this variable is set to TRUE, the **animal** has its red signal showing on its back.	Boolean	FALSE
blue.signal	When this variable is set to TRUE, the **animal** has its blue signal showing on its back.	Boolean	FALSE
body.size	Upon conception, **animals** have a body.size of 0.01, but this value can change over time depending on how much energy is allocated to growth.	Rational number from 0 to 1.0	0.01
body.shade	Upon conception, **animals** have a body.shade of 0, but this value can change over time depending on how much energy is allocated to *body-shade*.	Rational number from 0 to 1.0	0
is.resting	When this variable is set to TRUE, an **animal** cannot move or perform actions that cause movement.	Boolean	TRUE
identity.I	This variable represents a chromosome of letters that are inherited from its parents through recombination and mutation mechanisms during reproduction.	List	Inherited from parents
identity.II	This is the second chromosome of letters. These state variables are used in combination to calculate the degree of relatedness between individuals.	List	Inherited from parents
carried.items	A list of objects that an **animal** is currently carrying, which can be other animals and are typically gestates or infants.	List	Empty list
(f) Hidden attributes			
hidden.chance	The probability that an **animal** will become hidden from view (or unhidden) at that timestep.	Rational number from 0 to 1.0	0
fully.decayed	This variable is set to true when an **animal** has died and decayed beyond further recognition from other **animals**.	Boolean	FALSE
survival.chance	This sets the probability that the **animal** will survive to the next timestep.	Rational number from 0 to 1.0	1
energy.supply	This variable stores all energy reserves for **animals**, from which they can draw to pay for actions.	Positive rational number	0
bite.capacity	The amount of energy that can be consumed from each **plant** per timestep.	Rational number from 0 to 1.0	0.1
mutation.chance	During reproduction, alleles may mutate based on the average mutation.chance of the mating pair.	Rational number from 0 to 1.0	0.01
sex.ratio	The chance that a mother will conceive a male **animal**.	Rational number from 0 to 1.0	0.5
litter.size	The expected amount of offspring that a mother will conceive at one time.	Rational number from 0 to 1.0	0
conception.chance	During mating, **animals** have a chance of conception based on this value and their partner's value.	Rational number from 0 to 1.0	0
visual.angle	The angle that an **animal** can see at one time.	Rational number from 0 to 1.0	0
visual.range	How far away an animal can see, calculated as maximum-visual-range * visual.range.	Rational number from 0 to 1.0	0
day.perception	The percentage of objects seen during the day within the cone of perception created by the visual.angle and visual.range.	Rational number from 0 to 1.0	0
night.perception	The percentage of objects seen at night within the cone of perception created by the visual.angle and visual.range.	Rational number from 0 to 1.0	0

`yellow.chance`	The chance of the `yellow.signal` turning on (`True`) or off (`False`).	Rational number from 0 to 1.0	0
`red.chance`	The chance of the `red.signal` turning on (`True`) or off (`False`).	Rational number from 0 to 1.0	0
`blue.chance`	The chance of the `blue.signal` turning on (`True`) or off (`False`).	Rational number from 0 to 1.0	0
`birthing.chance`	The chance that a pregnant **`female`** will give birth at this time step.	Rational number from 0 to 1.0	0
`weaning.chance`	The chance that a lactating **`female`** will wean her dependent offspring at this time step.	Rational number from 0 to 1.0	0
`infancy.chance`	The chance that a gestatee will transition to an infant this time step.	Rational number from 0 to 1.0	0
`juvenility.chance`	The chance that an infant will transition to a juvenile at this time step.	Rational number from 0 to 1.0	0
`adulthood.chance`	The chance that a juvenile will transition to an adult at this time step.	Rational number from 0 to 1.0	0
`x.magnitude`	Desired magnitude of direction to take along the x axis, which when combined with the `y.magnitude` provides a trajectory.	Positive or negative rational number	0
`y.magnitude`	Desired magnitude of direction to take along the y axis, which when combined with the `x.magnitude` provides a trajectory.	Positive or negative rational number	0
`chromosome.I`	These chromosomes contain inherited behavioral weight preferences.	List	Inherited from parents through chromosomal recombination.
`chromosome.II`	These chromosomes contain inherited behavioral weight preferences.	List	Inherited from parents through chromosomal recombination.
`my.environment`	The list of environmental objects seen at this time step.	List	Empty list
`decision.vectors`	The list of decisions made this time step.	List	Empty list
`actions.completed`	The list of actions taken at this time step.	List	Empty list
(g) Tracking variables			
`age.in.ticks`	This variable tracks the current age measured in time steps.	Integer	0
`generation.number`	Tracks the number of generations that have been produced since the population was initialized.	Integer	Generation of mother + 1
`mother.identity`	After initialization, **`animals`** are conceived through sexual reproduction from a mother and father.	meta.id	meta.id of mother
`father.identity`	After initialization, **`animals`** are conceived through sexual reproduction from a mother and father.	meta.id	meta.id of father
`natal.group.id`	The identity of the group to which an **`animal`** was first assigned at conception, which is the same as their mother's.	meta.id	Assigned to same group as mother
`natal.group.size`	The total number of individuals, including themselves, in the group to which they are born.	Integer	Calculated number of group members
`ticks.at.conception`	The simulation timestep at which they were conceived, thus created as a new **`animal`** within the simulation.	Integer	Current timestep
`ticks.at.birth`	The simulation timestep at which an **`animal`** was born.	Integer	0
`ticks.at.weaning`	The simulation timestep when an **`animal`** is weaned and can no longer nurse from its mother.	Integer	0
`ticks.at.sexual.maturity`	The simulation timestep when an **`animal`** becomes an `adult` and is able to reproduce.	Integer	0
`ticks.at.death`	The simulation timestep when an **`animal`** dies and their body begins to decay.	Integer	0
`adult.hidden.chance`	The value of `hidden.chance` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.survival.chance`	The value of `survival.chance` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.body.size`	The value of `body.size` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0

`adult.body.shade`	The value of `body.shade` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.energy.supply`	The value of `energy.supply` recorded upon an **`animal`** reaching sexual maturity.	Rational positive number	0
`adult.bite.capacity`	The value of `bite.capacity` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.mutation.chance`	The value of `mutation.chance` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.sex.ratio`	The value of `sex.ratio` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.litter.size`	The value of `litter.size` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.conception.chance`	The value of `conception.chance` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.visual.angle`	The value of `visual.angle` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.visual.range`	The value of `visual.range` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.day.perception`	The value of `day.perception` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.night.perception`	The value of `night.perception` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.yellow.chance`	The value of `yellow.chance` is recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.red.chance`	The value of `red.chance` is recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`adult.blue.chance`	The value of `blue.chance` recorded upon an **`animal`** reaching sexual maturity.	Rational number from 0 to 1.0	0
`distance.traveled`	The distance in cell length that an **`animal`** has traveled up to that point.	Rational positive number	0
`mother.initiated.birth`	This value is `true` if the mother started the birthing process and `false` if the offspring initiated their own birth.	Boolean	FALSE
`mother.initiated.weaning`	This value is `true` if the mother started the weaning process (and stopped lactating) and `false` if the offspring initiated their own weaning.	Boolean	FALSE
`whole.related.help.cost`	The sum of all energy invested in *`helping`* individuals who are 100% related (including self).	Rational positive number or zero	0
`half.related.help.cost`	The sum of all energy invested in *`helping`* individuals who are 50% related (siblings, parents, and offspring).	Rational positive number or zero	0
`fourth.related.help.cost`	The sum of all energy invested in *`helping`* individuals who are 25% related (cousins, grandparents, and grand offspring).	Rational positive number or zero	0
`eighth.related.help.cost`	The sum of all energy invested in *`helping`* individuals who are 12.5% related (more distant relations).	Rational positive number or zero	0
`foraging.gains`	The total lifetime amount of energy acquired through foraging.	Rational positive number or zero	0
`total.energy.gains`	All energy that has been gained either by foraging or by nursing.	Rational positive number or zero	0
`total.energy.cost`	The sum of all energy that has been used to pay for actions.	Rational positive number or zero	0
`receiving.history`	The lifetime record of all individuals who have provided energy during nursing.	List	Empty list
`carried.history`	The lifetime record of all individuals who have been carried.	List	Empty list
`aid.history`	The lifetime record of all individuals who have been provided aid.	List	Empty list
`harm.history`	All individuals who an **`animal`** has harmed during their lifetime.	List	Empty list

`copulations.history`	All individuals who an **`animal`** has mated during their lifetime.	List	Empty list
`conceptions.history`	All individuals who an **`animal`** has mated and conceived with during their lifetime.	List	Empty list
`group.transfers.history`	All groups that an **`animal`** has transferred to during their lifetime.	List	Empty list
`infanticide.history`	All infants who an **`animal`** has harmed during their lifetime, enough such that they die before reaching juvenile period.	List	Empty list

The name of the state variable (a) and description of that attribute (b) in B3GET. (c) Presents a range of possible values for the state variable. (d) Indicates the values given to new **`animals`** upon conception. Visible attributes (e) are state variables that all **`animals`** can see while hidden attributes (f) are state variables that are hidden to all **`animals`** but the possessor of the attribute. Tracking variables (g) are state variables that are not accessible to any animal and instead track various life outcomes for user analysis.

3 Process overview and scheduling

B3GET is executed by the following processes: (1) the **`plant`** subroutines that determine growth at each timestep, (2) the **`animal`** subroutines that calculate and execute their decisions, and (3) environmental constraints imposed by the model. These processes are described in more detail in Submodels (7).

4 Design concepts

Basic principles. B3GET seeks to implement principles of behavioral ecology (Krebs and Davies, 1993), with concepts from virtual evolution (Dawkins, 1976; Holland, 1995; Neumann, 1966; Ray, 1991; Yaeger, 1993), and research in life history theory (Stearns, 1992), to simulate populations of virtual organisms evolving over generations, whose evolutionary outcomes reflect the selection pressures of their environment. I built B3GET with the assumption that, like all models, it exists as a separate universe coded with a combination of rules that define the physical realities of that universe (Hogeweg, 1989). This distinction is important so that a model is not confused with reality, but instead serves as a simplified proxy of that reality. The purpose of B3GET, which is to study how populations evolve in response to different ecological contexts, is not hindered by this assumption: as long as traits are inherited and correlate with fitness, selected behavioral strategies should emerge from any population, even a virtual one in a different ecological context of a virtual world. Thus, one can think of B3GET as a way to study the evolution of life contained in a virtual petri dish, or a miniature fieldsite (Premo, 2006).

Emergence. B3GET focuses on the emergence of behavioral strategies from a given ecological context over the course of generations of selection. The ecological context of **`animals`** includes both the surrounding plant ecology and the distribution and attributes of other **`animals`**. The plant ecology depends mainly on six parameter settings: ***plant-annual-cycle*** and ***plant-daily-cycle*** respectively set the year and day length in timesteps; ***plant-seasonality*** influences the annual extremes of the `penergy.supply` in plants; the ***plant-minimum-neighbors*** and ***plant-maximum-neighbors*** dictate whether plants prefer to be isolated or surrounded by other plants, thus influencing the degree of heterogeneity in plant distribution; and ***plant-quality*** sets the maximum `pterminal.energy` possible for plants (Table 4). I describe the relationship between these parameters and the emergent behavior of **`plants`** in more detail below.

While the parameter settings outlined above create fairly deterministic bounds for plant behavior, the **`animals`** themselves are free to evolve in an unbounded way. B3GET includes more behavioral possibilities than can be predicted before a simulation starts. In Crouse (2019), we calculated the number of possible combinations of behaviors based on the available options. These options have expanded in the current B3GET version, which corresponds to an updated calculation of over 300 trillion[1] for all possible combinations of behaviors. Like in the previous version, this calculation does not consider differences in the numerical values contained in most alleles, which are freely able to evolve to any rational number between 0 and 1, and thus the actual possible combination of behaviors incorporating this component is infinite. Evolved strategies include the timing of life history events like gestation and weaning; the amount of energy allocated to body growth and maintenance; amount of time spent foraging; and others. Furthermore, these behaviors include movement in any direction on the 2-dimensional landscape, based on their genetically-derived weighted preferences for moving relative to other **`animals`**. Thus, population level spatial dynamics, and other emergent patterns, are not imposed by the model but emerge from individual interactions and movement preferences, which evolve over generations.

---

[1] There are 2,654,208 possible status options for **`animals`**, 40 status options for **`plants`**, and 43 possible actions. Each behavior combination is based on the unique combination of actor's (**`animal`**) status, target's status (**`animal`** or **`plant`**), and action type, resulting in 302,932,829,850,112 possible combinations, compared to 4,285,540,224 combinations calculated for BEGET in Crouse et al. (2019).

Adaptation. `Animal` populations can evolve through the inheritance of imperfectly-copied alleles during sexual reproduction, selection, genetic drift, and if different the user combines multiple **`animal`** populations, gene flow. I represent inheritance using a set of diploid chromosomes that together form an **`animal's`** genotype. Initial populations of **`animals`** are given genotypes through information that is stored in external files. Each chromosome consists of a set of genes which encode rules for which behaviors to perform depending on the environmental context (see Submodels for more detail). Each allele contains a list of codons, which are represented in this model as either letters or a number, whose combinations shape the behavior of an **`animal`** within a given context. Considering that these genes can include numbers representing a relative desire to perform a behavior, each gene can have an indefinite number of possible alleles.

`Animals` reproduce sexually and so prior to conception, chromosomes undergo the genetic processes of recombination and mutation. These alleles are organized in such a way to allow for these processes, thus enabling populations to evolve novel behaviors over generations. During recombination, B3GET randomly selects one of two chromosomes from each parent. Then, B3GET randomly exchanges roughly 50% of alleles from the selected chromosome with alleles from the homologous chromosome. Finally, each parent provides this chromosome to produce a new homologous pair of chromosomes in the offspring. After this process, mutation at each locus occurs by chance based on the average `mutation.chance` of the offspring's parents. Should a mutation event occur at a locus, one of the following processes is selected at random to happen: (1) the allele is deleted, (2) the allele is duplicated, (3) the allele is modified by (i) deleting a codon, (ii) duplicating a codon, or (iii) modifying a codon with a new value. The first codon of each allele is immutable and determines if only a subset of the previously described mutation processes are allowed or if no mutation is allowed at this allele. These options are set by the user when a simulation is initialized and do not change as the simulation progresses. In this way, users can influence which behaviors can evolve and which cannot, which offers a high level of control for a variety of experiments.

Thus as a population of **`animals`** evolves, behavioral traits change over time due to variation in alleles resulting from mutation, new combinations of alleles resulting from recombination and sexual reproduction, and changes to the frequencies of alleles resulting from selection and drift.

Over generations, a population of **`animals`** is expected to evolve new strategies that allow them to achieve better fitness for the given environmental context, and individuals within the population to develop new strategies through mutation and chance that may lead to increased fitness and thereby propagate within the population.

Objectives. B3GET is designed with an environment that acts to select for individuals who are best able to survive and reproduce within the environment. At each timestep, B3GET imposes a fitness rule as follows. First, B3GET checks the `survival.chance` of each **`animal`** against a randomly generated number between 0 and 1. If the number falls above the `survival.chance`, which is also constrained to values between 0 and 1, then B3GET sets the `is.alive` state variable of the **`animal`** to "`false`," and the **`animal`** is no longer able to act. Such "dead" **`animals`** continue to decay according to the global parameter ***deterioration-rate*** and will eventually decay beyond recognition when it fails the check described above for a second time. Once an **`animal`** fails the check a second time, its body becomes invisible to other **`animals`** and is no longer able to be eaten. Thus an **`animal's`** survival depends solely on keeping its `survival.chance` high enough to pass this check at every timestep. Increasing the `survival.chance` value costs energy, as do all other behaviors related to mating and reproduction. Since an **`animal's`** fitness depends solely on surviving long enough to differentially reproduce more than its competitors, presumably the best behavioral strategies for a given environmental context should be selected.

Sensing. To simulate the limited visual fields of real animals, an **`animal`** can sense the surroundings that are in its "cone of perception" (Figure 2a), which is calculated based on its `visual.range` and `visual.angle`. How well an **`animal`** can sense other objects within its cone of perception is determined by its `day.perception` during the day and `night.perception` at night. If an **`animal`** senses another object, then it has access to information about the visible attributes of that object. For **`plants`**, this means that **`animals`** can see their `pcolor`: a darker color indicates higher `penergy.supply`, which can influence whether an **`animal`** chooses to move toward the **`plant`** and eat it. **`Animals`** can also see the visible attributes of other **`animals`**, including their `biological.sex`, `body.size`, `body.shade`, current `life.history` stage, current `female.fertility` status, current `group.identity`, whether they are `is.alive` or `is.resting`, if they have their `yellow.signal`, `red.signal`, or `blue.signal` turned on, and if they have any `carried.items`. Additionally, **`animals`** can assess their degree of relatedness to others that they can see within their visual range.

Table 2. List of `animal` actions in B3GET

B3GET subroutine name (a)	Description of subroutine (b)	Positive effect (c)	Negative effect (d)
Intra-actions (e)			
survival-chance	Maintain body to decrease chance of dying.	Increase living.chance organ.	Decrease living.chance organ.
body-size	Change body size.	Increase body.size organ.	Decrease body.size organ.
body-shade	Change body shade from lighter to darker.	body.shade increases	body.shade decreases
day-perception	Change range of perception of other animals during daytime.	day.perception.range increases	day.perception.range decreases
night-perception	Change range of perception of other animals during nighttime.	night.perception.range increases	night.perception.range decreases
visual-angle	Change daytime perception angle.	day.perception.angle increases	day.perception.angle decreases
visual-range	Change nighttime perception angle.	night.perception.angle increases	night.perception.angle decreases
conception-chance	Change chance to conceive during a mating event.	conception.chance increases	conception.chance decreases
bite-capacity	Change the stomach capacity.	stomach.size increases	stomach.size decreases
mutation-rate	Change chance of mutations occuring during reproduction.	mutation.chance increases	mutation.chance decreases
sex-ratio	Change sex ratio of next litter of offspring.	sex.ratio increases	sex.ratio decreases
litter-size	Change the amount of offspring in the next litter.	litter.size increases	litter.size decreases
turn-right	Rotate to the right.		turn-left
turn-right (-)	Rotate to the left.	turn-left	turn-right
go-forward	Go forward.		
hide			
yellow-signal	Change the chance that the alpha signal is displayed.	Increase alpha.chance	alpha.chance decreases
red-signal	Change the chance that the beta signal is displayed.	beta.chance increases	beta.chance decreases

blue-signal	Change the chance that the gamma signal is displayed.	gamma.chance increases	gamma.chance decreases
check-infancy	Checks whether a gestatee becomes an infant.	infancy.chance increases	infancy.chance decreases
check-birth	Checks whether pregnant female gives birth to her dependent gestatee.	birthing.chance increases	birthing.chance decreases
check-juvenility	Checks whether an infant becomes a juvenile.	juvenility.chance increases	juvenility.chance decreases
check-weaning	Checks whether lactating female weans her dependent infant.	weaning.chance increases	weaning.chance decreases
check-adulthood	Checks whether a juvenile becomes an adult.	adulthood.chance increases	adulthood.chance decreases
Inter-actions (f)			
move-toward	Turn toward target.	Turn toward target.	Turn away from target.
move-toward (-)	Turn away from target.	Turn away from target.	Turn toward target.
supply-to	Make available energy supply for target.	Insist on giving energy to target.	Resist giving energy to target.
demand-from		Insist on taking energy from target.	Resist taking energy from target.
eat	Ingest some of the energy supply of target.	Take energy from target.	—
join-group-of	Join group of target	Insist on joining same group as target.	Resist target joining group or insist that target leaves group.
join-group-of (-)	Leave group of target (if target is in current group)	Insist on leaving same group as target.	Resist target leaving group or insist that target joins group.
recruit	Make other individual join my group.		
expel	Make other individual leave my group.		
pick-up	Pick up target and put in inventory.	Insist on carrying target.	Insist on not carrying target.
pick-up (-)	Put down target, take out of inventory of carried items.	Insist on not carrying target.	Insist on carrying target.
cling-to		Insist on being carried by target.	Resist being carried by target.
cling-to (-)	Opposite effect of cling-to.	Resist being carried by target.	Insist on being carried by target.
help	If within the same cell, groom Target to decrease their mortality-chance.	Insist on helping target.	Resist being attacked by target.
attack	If within the same cell, calculate likelihood of winning fight based	Insist on attacking target.	Resist being helped by target.

	on relative body-size, loser increases mortality-chance.		
mate-with	Both agents negotiate whether a mating occurs, and if so, there is a chance of conception determined by the average of their `conception.chance` organ size.	Insist on mating with target.	Resist mating with target.

Self-directed actions involve internal mechanisms that may change the state variables of the actor. Target-directed actions involve interactions between the actor and a target in their environment that influence the actor's movement and may change the state variables of either the actor or target. Actions have corresponding genotype codes, which are upper-case letters encoded into genotype files to indicate the action to take for each gene accessed. Intra-actions are allowed to be positive and negative. (a) The name of the corresponding subroutine in B3GET code, which is generally the action name itself, or in some cases the reciprocal of another action and so is equivalent to a negative number passed. (b) Description of the subroutine code in lay terms. (c) Describes the specific effect of passing a negative number as the energy spent. (d) Specific effect of passing a negative number. (e) Intra-actions are actions that are directed at the self. (f) Inter-actions are actions that are directed at another target and only trigger if the target is spatially adjacent to self.

Interaction. An **`animal`** performs behaviors to modify its own attributes or the attributes of others. An **`animal`** can engage in any number of behaviors per timestep, depending on its current `energy.supply` available to use, and its decisions generated from the environmental context and its genotype. I call behaviors that modify an **`animal's`** own state variables intra-actions, and inter-actions are those that modify the state variables of other **`animals`**. For example, an **`animal`** can invest energy to increase its own `survival.chance` or pay energy to *`attack`* another **`animal`**, thus decreasing their victim's `survival.chance`. Table 2 includes a full list of available actions.

An **`animal`** can perform intra-actions to change its own state variables, many of which alter the likelihood of a physiological event occurring. For example, **`animals`** use the *`infancy-chance`* action to increase their `infancy.chance`, which eventually triggers a `gestatee` to transition into an `infant`. Likewise, `"pregnant"` **`females`** can invest in `birthing.chance` to trigger birth and `"lactating"` **`females`** can invest in `weaning.chance` to restart their cycling and stop nursing their dependent offspring. **`Animals`** must invest in their `survival.chance` to avoid death, and in their `conception.chance` to conceive during mating. Other intra-actions can alter attributes related to physiological abilities, such as `visual.range` and `visual.angle`, which determine how big is their ‘cone of perception’ (see Figure 2).

`Animals` can interact with others to alter their state variables, or to reproduce. **`Animals`** interact with other **`animals`** in the following ways: *`attack`*, *`help`*, *`mate-with`*, *`join`*, *`leave`*, *`recruit`*, *`expel`*, *`pick-up`*, *`put-down`*, *`cling-to`*, and *`squirm-from`*. These interactions can only occur if the distance between both parties is less than half of the sum of their diameters (i.e., they occupy overlapping space). The *`attack`* and *`help`* interactions result in either a positive or negative change for the `survival.chance` of the receiver. When an **`animal`** attempts to *`mate-with`* another **`animal`**, it may cause conception to occur, depending on the mean `conception.chance` and cooperation of both parties, and provided that they do not share the same `biological.sex`. Conception results in the creation of a new **`animal`** whose alleles are copies from both parents, half from each. The *`join`*, *`leave`*, *`recruit`*, *`expel`* actions relate to altering the `group.identity` of an individual, which alters to which group of individuals they are affiliated. The *`pick-up`*, *`put-down`*, *`cling-to`*, and *`squirm-from`* actions relate to altering the `carried.items` of an individual. For example, a mother can *`pick-up`* and *`put-down`* her infant, and likewise an infant can *`cling-to`* and *`squirm-from`* her mother. When an **`animal`** carries another **`animal`**, the carrier must energetically compensate for the extra weight.

Like biological organisms, **`animals`** are also expected to indirectly interact through feeding and mating competition.

Table 3. Summary of limitations on `animals` according to their status

Life history status (a)	Female fertility status (b)	Description	Able to see and be seen by non-maternal others? (c)	Able to perform general intra-actions? (d)	Able to perform specialized intra-actions? (e)	Able to perform general inter-actions? (f)	Able to perform specialized inter-actions? (g)
gestatee	—	An **`animal`** who has just been created through	no	yes except *go-forward*	*`check-infancy`*, *`check-juv`*	no	*`demand-from`*

		conceive-with.			*enility*, *check-adulthood*		
infant	—	An **animal** whose mother has given birth to but has not weaned them.	yes	yes	*check-juvenility*, *check-adulthood*	yes except *eat*	*demand-from*
juvenile	—	An **animal** who is weaned but not yet adult (Pereira & Altmann, 1985).	yes	yes	*check-adulthood*	yes	no
adult male	—	An **animal** who has reached sexual maturity. Males and cycling females are able to *conceive-with* each other.	yes	yes	no	yes	*mate-with*
adult female	cycling		yes	yes	no	yes	*mate-with*
	pregnant	An **animal** who currently has at least one dependent gestatee.	yes	yes	*check-birth*	yes	*supply-to*
	lactating	An **animal** who currently has at least one dependent infant.	yes	yes	*check-weaning*	yes	*supply-to*
dead	—	An **animal** who no longer `is.alive` but their body still remains in the simulation.	yes	no	no	no	no

Animals are constrained in what actions they can perform based on their current `life.history`, `biological.sex` and `female.fertility` status. (a) **animals** have one of the following `life.history` status: gestatee, infant, juvenile, or adult. **animals** who are dead are no longer able to perform any actions. (b) **animals** who are `adult` and `female` have one of the following `female.fertility` status: cycling, pregnant, and lactating. (c) They are further limited in who they can see in their environment: only mothers are able to see their `gestatee` offspring. (d) Intra-actions are self-directed actions and the general intra-actions all living **animals** can perform include *survival-chance*, *body-size*, *body-shade*, *day-perception*, *night-perception*, *visual-angle*, *visual-range*, *conception-chance*, *bite-capacity*, *mutation-rate*, *sex-ratio*, *litter-size*, *turn-right*, *turn-left*, *go-forward*, *yellow-signal*, *blue-signal*, *red-signal*. (e) Specialized intra-actions include *check-infancy*, *check-birth*, *check-juvenility*, *check-weaning*, *check-adulthood* and can only be performed to completion by the statuses indicated (for example, only `pregnant` **females** who *check-birth* can actually give birth). (f) Inter-actions include all target-directed actions, or actions that target other **animals**: general actions include *move-toward*, *eat*, *join*, *leave*, *recruit*, *expel*, *pick-up*, *put-down*, *cling-to*, *squirm-from*, *attack*, *help*. (g) Specialized inter-actions include *supply-to*, *demand-from*, *mate-with* and can only be performed to completion by the statuses indicated (for example, only adult **males** and cycling **females** who *mate-with* each other can conceive).

Stochasticity. B3GET includes several stochastic processes in an analogous way to the natural processes affecting living biological organisms. At each timestep, B3GET chooses **plants** and **animals** in a random order to perform their subroutines. The calculations for **plant** growth relate to the likelihood of growth occurring. The genetic processes of recombination and mutation are also stochastic. Additionally, as I described above, many **animal** attributes are defined as numbers between 0 and 1 to represent the chance of an event occurring, which are all calculated stochastically. For example, the survival.chance attribute directly corresponds to the chance of surviving each timestep. Other attributes contribute to stochastic events more indirectly. For example, **animals** with a larger body.size have a proportionally greater chance of landing a blow during an *attack*.

Collectives. Each **animal** has a group.identity, which identifies to which group they belong. Often multiple **animals** belong to the same group and will spatially aggregate with each other based on their shared group membership. However behavioral preferences related to group composition and affiliation can be allowed to evolve freely. In fact, **animals** could evolve to act without any consideration of their group.identity at all.

Observation. B3GET can record a rich array of information, including (1) population files, (2) genotype files, (3) meta information, and (4) measurements of the simulation output. Population and genotype files are created by recording the instantaneous state of every individual, containing complete information that can be used as the starting conditions for a population of **animals**. For a new simulation to be initialized, B3GET must reference a population file, which provides the information needed to set up the initial population. Recording the exact state of a population allows users to run additional experiments using this information. B3GET also generates some behind-the-scenes files, including a "meta" file that contains user mouse-click information and a simulations file that contains a list of all simulation runs performed. Finally, B3GET can record detailed measurements of various emergent phenomena from a simulation. Users can adjust the settings to record information relevant to tracking the unfolding of major life events, such as when an animal reaches sexual maturity or when they conceive, and population-level information, such as demographic information and population averages for attributes like `body.size` and `conception.chance`. Some measurements are taken only once, when a simulation runs or a life has completed, which includes information that summarizes the simulation or life that has transpired. Other information is recorded at predetermined intervals to track how values change over time. The B3GET user manual includes more detailed information on how the user can adjust settings to generate data.

5 Initialization

Upon initialization, B3GET populates the world with **`animals`** and **`plants`**. One **`plant`** is placed in each cell of the 100 x 100 2-dimensional toroidal landscape and begins with a random starting `penergy.supply`. Animals are placed and defined according to the population file used to initialize the simulation. To generate an initial population of **`animals`**, B3GET requires access to a population csv file from which to obtain the necessary information. Access to a genotype file may also be necessary if that information does not already exist within the population file.

Users must make choices in the global parameter settings during the initialization process, which affects the emergent properties of the simulation (Table 4). There are two parameters that relate to record-keeping and do not otherwise affect the simulation. The ***model-version*** keeps a record of the current version of B3GET, which is recorded during the data collection process to keep track of the specific code used during the simulation. Likewise, ***simulation-id*** keeps track of the unique identifier of the current simulation, which is also recorded during data collection to track which results are related to which simulations.

Table 4. B3GET input parameter settings and baseline values

Parameter	Description	Value(s) used for this thesis
model-version	The version(s) of B3GET used for this experiment.	1.1.0

model-structure	The name of the B3GET subroutines used in a simulation	–
genotype-reader	The name of the B3GET subroutine that translates genotypes into decision-vectors.	“sta2us”
simulation-id	The unique identification code of a simulation.	–
simulation-stop-at	User setting to define when to end the simulation	100,000
Environment settings		
plant-annual-cycle	The length of a year in timesteps in the simulation.	1000
plant-daily-cycle	The length of a day in timesteps.	10
plant-seasonality	The degree of difference in plant abundance from summer to winter.	0.5
plant-quality	The maximum energy that any **`plant`** can contain.	4
plant-min-neighbors	The preferred minimum number of neighbors for each **`plant`**.	0-7
plant-max-neighbors	The preferred maximum number of neighbors for each **`plant`**.	1-8
deterioration-rate	The rate of decay in the simulation, which affects the `survival.chance` of **`animals.`**	-0.01
maximum-visual-range	Maximum possible range for an **`animal's`** cone of perception.	5 cells
base-litter-size	The maximum number of offspring per litter that **`animals`** can evolve to have.	10
maximum-population-size	The maximum population size when the "reaper" ***model-structure*** code is activated.	200
minimum-population-size	The minimum population size when the "stork" ***model-structure*** code is activated.	200
solar-status	Whether it is currently day or night in the simulation.	(d)
current-season	Current time of year in the simulation.	(d)
Results settings		
timestep-interval	Period between general simulation records	2000
simulation-summary-ticks	Timestep when summary information about the simulation is recorded	25,000
simulation-scan-ticks	Period between scans of the current simulation	250
group-scan-ticks	Period between scans of all groups in the current simulation	2000
individual-scan-ticks	Period between scans of all living individuals in current simulation	0
view-scan-ticks	Period between scans of the simulation view display	10,000
genotype-scan-ticks	Period between scans of the population average genotype	10,000
focal-follow-rate	Rate at which an individual is chosen to follow	0.0001
record-individuals	When true, records information on all anima1s who die	“true”

verification-rate	Rate at which the simulation is assessed	0.00001
record-world-ticks	Timestep when the NetLogo world is exported	100,000
start-date-and-time	The date and time when the simulation was first setup	–
plant-abundance-record	A periodic record of the plant abundance in the simulation	–
plant-patchiness-record	A periodic record of the plant patchiness in the simulation	–
population-size-record	A periodic record of the population size in the simulation	–
decisions-made-this-timestep	Complete list of decisions made by all **`animals`** this timestep	–
actions-completed-this-timestep	Complete list of actions completed by all **`animals`** this timestep	–
verification-results	Record of verification assessments during this simulation	–

The name and description of all B3GET input parameters and a summary of the values used for this experiment in this thesis.

There are two parameters that relate to which parts of the model code are used in the current simulation. B3GET is designed with a modular structure that allows parts of the code to be turned on or off to affect the overall functionality of the simulation. The ***model-structure*** setting allows the user to choose the overall structure of the model and includes the following options: `“baseline”` keeps the code structure to the original settings; `“no-plants"` removes the plant growth functionality and is mainly used to make it easier to see the **`animals`** during visual validation testing; `"reaper"` sets an artificial carrying capacity of 100 **`animals`** in the simulation and randomly removes **`animals`** that exceed this limit; conversely, `"stork"` ensures that **`animals`** randomly reproduce if the population size dips below 100 individuals; `"aspatial"` removes all spatial representation from the model and instead allows **`animals`** to interact with any other individual present without a constraint of locality; `"free-lunch"` enables **`animals`** to perform actions even if they do not have the required energy to pay for it; `"no-evolution"` removes the recombination functionality during reproduction and instead individuals receive their entire genotype from their mother with no option for mutation of alleles; `"ideal-form"` enables all newly conceived individuals to assume the average value for all phenotype variables; and `"uninvadable"` allows for recombination during reproduction but disables mutation. The ***genotype-reader*** also controls the code that is specifically related to translating the genotypes of **`animals`** into actions for that **`animal`** to take: "`sta2us`" is a modified version of the original code used in Crouse et al. (2019), and "`gat3s`" is a new sequence of code designed to handle more sophisticated genotype configurations. Both extensions come with a corresponding genotype file, which is given to all individuals initialized at the start of a simulation.

All other global parameters relate to shaping the emergent properties of the environment in which the **`animals`** live. The ***deterioration-rate*** sets the rate of artificial decay accumulated by **`animals`** in a simulation; the higher this setting, the more likely **`animals`** are to die and the more necessary it becomes for them to invest in increasing their `living.chance`. This artificial decay rate also sets the rate at which dead **`animal`** bodies decay beyond recognition by other living **`animals`**. The remaining environmental parameters collectively determine the patterns of plant growth and decay over time. The ***plant-annual-cycle*** determines how long a year lasts during a simulation and by default is set to 1000 timesteps. The ***plant-daily-cycle*** determines how many timesteps represent a single day; therefore dividing the former value by the latter value determines how many days there are in a simulated year. For example, the default setting is 10 timesteps per day, resulting in 100 days per year. Over the course of one daily cycle, the simulation transitions between day and night settings, which can alter an **`animal's`** ability to perceive their surroundings depending on their perception attributes related to day and night. ***plant-seasonality*** determines how extreme the environment changes are over an annual cycle, with 0 meaning no change throughout the year to 1 meaning an extreme change from one half of the year to the other, simulating extreme fluctuations between summer and winter, or wet and dry seasons. ***Plant-quality*** determines the maximum amount of energy a **`plant`** can possess, which can be considered the terminal length of a plant. Two parameter settings, ***plant-minimum-neighbors*** and ***plant-maximum-neighbors*** represent the lower bound and upper bound of plant preference for the number of plant neighbors they have adjacent to them. Collectively, these environmental settings combine to determine at each timestep how much energy is gained or lost for each plant in the simulation.

6 Input data

B3GET initially creates a collection of **`animals`** from a population file and gives these **`animals`** an initial genotype copied from a genotype file. If a genotype file is not indicated in the genotype input parameter, then B3GET attempts to find information about individual genotypes directly from the population file, if this information exists. B3GET requires an initial population to begin any simulation, and an initial genotype file, if the population file does not provide this information. These files can be modified to create a range of initial conditions for a seed population. Seed populations are populations that initially populate the virtual B3GET environment at the beginning of a simulation, from which new populations "grow" through evolutionary processes as the simulation runs. These populations are stored in csv files that contain information for every state variable of every **`animal`** in the population. When B3GET imports a population file, it creates a set of **`animals`** that exactly matches the information that is stored in the file, including their spatial location, heading, and attributes. Within these files, each row represents an **`animal`** and each column represents one of its state variables; the headings in the first row store the state variable names and the cells below each header store the value of that state variable for the **`animal`** in the corresponding row.

Each row in the genotype file represents a gene locus for a particular operation and the quantitative values in these rows represent allele variants for this gene. The first codon of each allele is immutable and does not affect behavior, but instead enables the user to define the type of mutation possible at that locus during reproduction. The possible types of mutation include (1) deleting an allele, (2) duplicating an allele, (3) modifying an allele by (i) deleting a codon, (ii) duplicating a codon, and (iii) modifying a codon with a new value.

7 Submodels

Conway's Plants.

I constructed B3GET to include a wide range of **`plant`** behaviors to better understand how environmental factors shape evolved behaviors. I modeled **`plant`** dynamics based on the rules of Conway's Game of Life, a cellular automaton model that simulates a changing ecology of 'alive' or 'dead' cells (Gardner, 1970). These rules define the state of a cell according to the states of its eight neighboring cells: an alive cell continues living if they are next to two or three living neighbors; a dead cell becomes alive if they are next to exactly three neighbors; and all other cells become dead or remain dead. From these simple rules emerges complex shapes and patterns that resemble the unfolding of life and complexity observed in the natural world. In B3GET, instead of existing in a binary state, **`plants`** have the variable attribute `penergy.supply`. The number of neighbors for a plant corresponds with the amount of `penergy.supply` that each neighbor contains, where a fully 'alive' **`plant`** would contain the maximum amount of energy possible, and a 'dead' **`plant`** contains no energy. Two parameter settings, ***plant-minimum-neighbors*** and ***plant-maximum-neighbors*** represent the lower bound and upper bound of plant preference for the number of plant neighbors they have adjacent to them. A **`plant`** grows more energy when surrounded by a preferred number of neighbors, and loses energy otherwise.

I desired to have plants that are constrained by user-selected parameters so that the environmental aspects could be controlled by users while the **`animal`** behaviors were allowed to evolve freely. To better trace the connection between environment and behavior, I sought a minimal set of parameters to fully describe the main ways in which plant behavior varies: ***plant-annual-cycle*** ($\mathbf{p_a}$), ***plant-daily-cycle*** ($\mathbf{p_d}$), ***plant-seasonality*** ($\mathbf{p_s}$), ***plant-quality*** ($\mathbf{p_q}$), ***plant-minimum-neighbors*** ($\mathbf{p_{min}}$), ***plant-maximum-neighbors*** ($\mathbf{p_{max}}$) (see Table 4 for descriptions of these variables). I use the variable designations in the parentheses for brevity in this section. Overall, these parameters are used to calculate how much growth is gained or lost by each **`plant`** at each timestep. In constructing the behavioral rules of plant growth, I applied the following criteria. First, provided that $\mathbf{p_s}$ is nonzero, **`plants`** are more abundant in the "high growth" season (i.e., summer or wet season) and less abundant in the "low growth" season (i.e., winter or dry season) and cycle between these extremes once every span of $\mathbf{p_a}$ (a year). Second, the growth of **`plants`** scales with both the quality of plants ($\mathbf{p_q}$), which modulates the maximum amount of energy that each plant can contain, and seasonality ($\mathbf{p_s}$), which modulates the positive or negative impact of the current season on growth. Third, $\mathbf{p_{min}}$ and $\mathbf{p_{max}}$ impact the pattern of plant distribution, allowing for scenarios in which **`plants`** are completely homogenous or extremely patchy. Fourth, **`plants`** are not stable at extreme conditions such as when they are all completely dead or all completely grown. I describe in more detail how I calculate plant growth using Equations 1 - 6 below.

The current season (***s***) in B3GET is defined such that it is -1 at the winter solstice and +1 at the summer solstice and sinusoidally cycles between solstices once every $\boldsymbol{p_a}$ timesteps. Thus, if $\boldsymbol{p_a}$ is set to a high number, like 100,000 timesteps, the season will cycle between -1 and +1 only once per 100,000 timesteps.

$$(1) \qquad s = cos\left(\frac{360}{p_a} \cdot timesteps\right)[-1.0,\ 1.0]$$

The current season (***s***) is modulated by the seasonality parameter ($\boldsymbol{p_s}$) to determine the actual seasonal impact ($\boldsymbol{I_s}$) on plant growth.

$$(2) \qquad I_s = \frac{1}{2}(s \cdot p_s + 1)\ [0, 1.0]$$

The impact of neighboring plants (I_n) specify the impact of a **`plant's`** immediate surrounding neighbors on its growth. I define two values, $\mathbf{p_{min}}$ and $\mathbf{p_{max}}$, that set the bounds between which plants thrive and outside of which they feel stifled and lose energy. Surrounding neighbors ($\mathbf{n_i}$) of the ith plant, which can be represented as fractions of a whole plant based on its current `penergy.supply` with respect to the maximum possible energy supply set by the parameter $\mathbf{p_q}$. The **`plants`** more securely within the bounds thrive the most and those well outside the bounds thrive the least. I use a Gaussian function to define a bell curve representing this relationship. Therefore the most optimal neighbor preference (***o***) would be equidistant between both parameters.

$$(3) \qquad o = \left|\frac{p_{min}+p_{max}}{2}\right|\ [0,\ 8.0]$$

Equation (4) combines the above terms into a Gaussian function, rearranged to be a rational number between -1 and +1.

$$(4) \qquad I_n = 2e^{\left(-\frac{(n-o)^2}{2(o-p_{min})^2}\right)} - 1\ \ [-1.0, 1.0]$$

Baseline change in energy growth (Δe_b) is how much a plant would grow at each timestep, removed from seasonal and other contextual factors.

$$(5) \qquad \Delta e_b = \frac{d \cdot q}{a}$$

I calculated **`plant`** density by considering the sum of all `penergy.supply` of patches. Impact of **`plant`** density (I_p) is a ratio of the number of plant units, defined by energy, divided by the number of cells in the world, which is 10,000 for the simulations in this thesis. Thus, the maximum change in energy from previous timestep (Δe_i) is calculated as the baseline change multiplied by the positive and negative impacts on plant growth for a given plant at a given timestep and surrounding context.

$$(6) \qquad \Delta e_i = (\Delta e_b)(I_p \cdot I_n - I_s + I_p)$$

To avoid deterministic effects, the final growth is calculated by randomly selecting from Δe_m. If ***model-structure*** includes "no-plants" then no plant growth occurs.

`Animal` behaviors. At each timestep, B3GET selects **`animals`** in a random order to decide which behaviors that they will perform (Figure 2). This process can be subdivided into three main parts: (1) consideration of the current environment, (2) calculation of decision-vectors based on genetically determined predispositions for the environment, and (3) execution of these decisions with an associated energetic cost. Below, I describe this process in greater detail.

First, an **`animal`** evaluates its environment, which includes individuals within its cone-of-perception, including itself. One exception is that only mothers are able to perceive their "`gestatee`" offspring and gestatees are only able to perceive themselves and their mothers. Individuals can also sense the individuals that are carrying them, and the individuals that they are carrying. Within this visible environment, an animal can see the visible attributes of other animals and plants, such as their `body.size` and `penergy.supply`, respectively.

Next, B3GET combines each **`animal's`** genotypes and current environmental context into a list of decision-vectors. We can consider each attribute that an animal can see a 'key' that can be used to 'unlock' one or multiple of its alleles. Alleles contain references to a certain environmental context, which includes a hypothetical attribute that an animal might see in its surroundings. We can consider this environmental context within the allele to be the ‘lock’ that can be unlocked from a matching ‘key’ found in the environment. During the decision-making process, an animal checks all of the keys it can see against each of its genes' locks. A key matches a lock if each element in the lock can be found in the key. If a match is found, then the remaining elements of the allele, which contain directions for performing actions, are put into a queue that contains all decision-vectors. See Figure 2 for an example of this process.

Finally, an **`animal`** executes all actions in its decision-vector queue, allocating a portion of its `energy.supply` to pay for the ability to perform the action. An action can only be executed if there is sufficient energy available to pay for that action. If an **`animal's`** `energy.supply` reaches 0 before it executes all actions in its decision-vector queue, then the remaining actions are not performed and the queue is cleared in the next timestep. Multiple actions can be executed within the same time step and many actions result in state variable changes (Table 2). Within B3GET, I call this process of gathering environmental information, and generating and executing actions to be the elements that comprise a behavior.

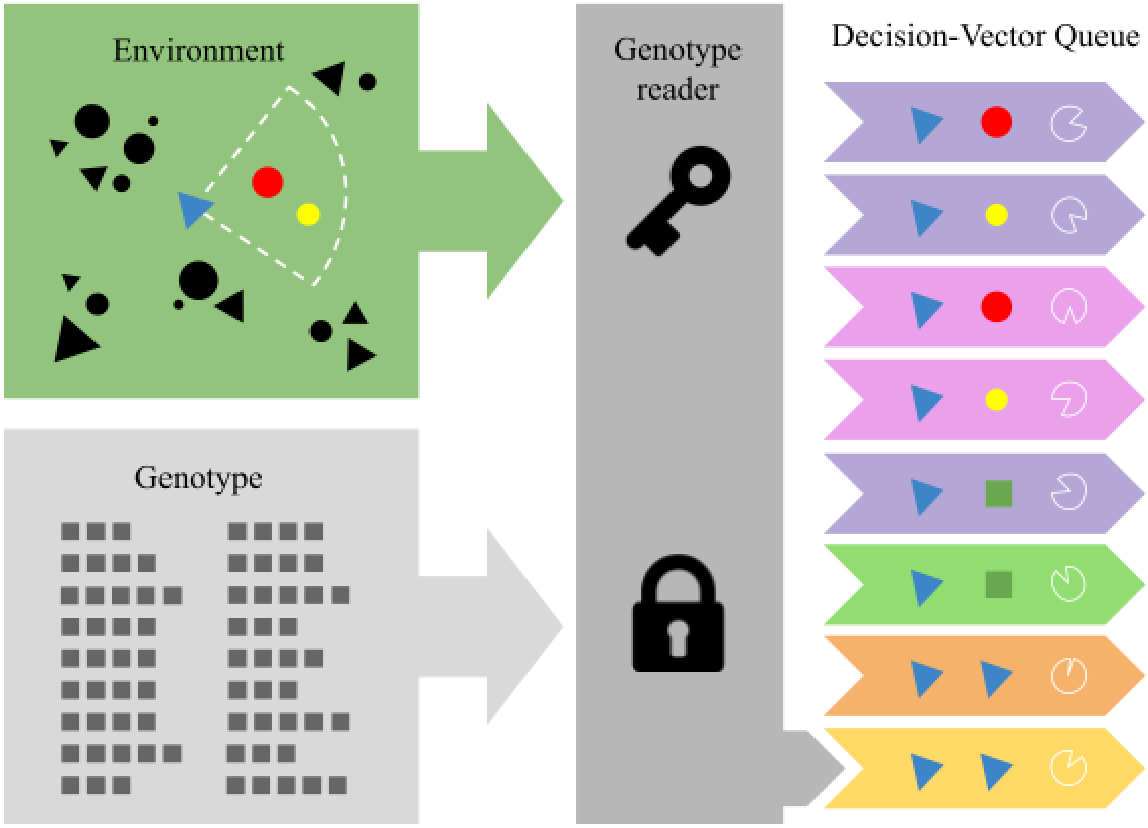

Figure 2 (a). Summary of the `animal` decision-making process depicting an arbitrary environmental context and genotype. The genotype contains two chromosomes depicted as two columns, containing rows of alleles, which are lists of codons. When **`animals`** make decisions, they take in information from their environment and what they can "see" is their cone of perception (dashed line) whose area is based on their perception attributes. The information from the environment can be thought of as keys and the genotype alleles can be thought of as locks. The "Genotype reader" code cross-references the keys and locks and if a match is found then it "unlocks" the allele and generates a decision-vector containing four elements: (1) the individual performing the action, (2) the individual being targeted with the action, (3) what kind of action is being performed, and (4) how much energy to allocate to the action. For example, an `adult` **`male`** (center triangle) sees two `adult` **`females`** (circles in field of vision) in his field of vision and makes decisions to *`move-toward`* and *`mate-with`* the females, as well as *`eat`* a nearby **`plant`** and perform internal functions such as increasing his `survival.chance` and `body.shade`. After the decisions are made and put in a queue of decision-vectors, the male will systematically perform each action, as long as he has enough energy to spend.

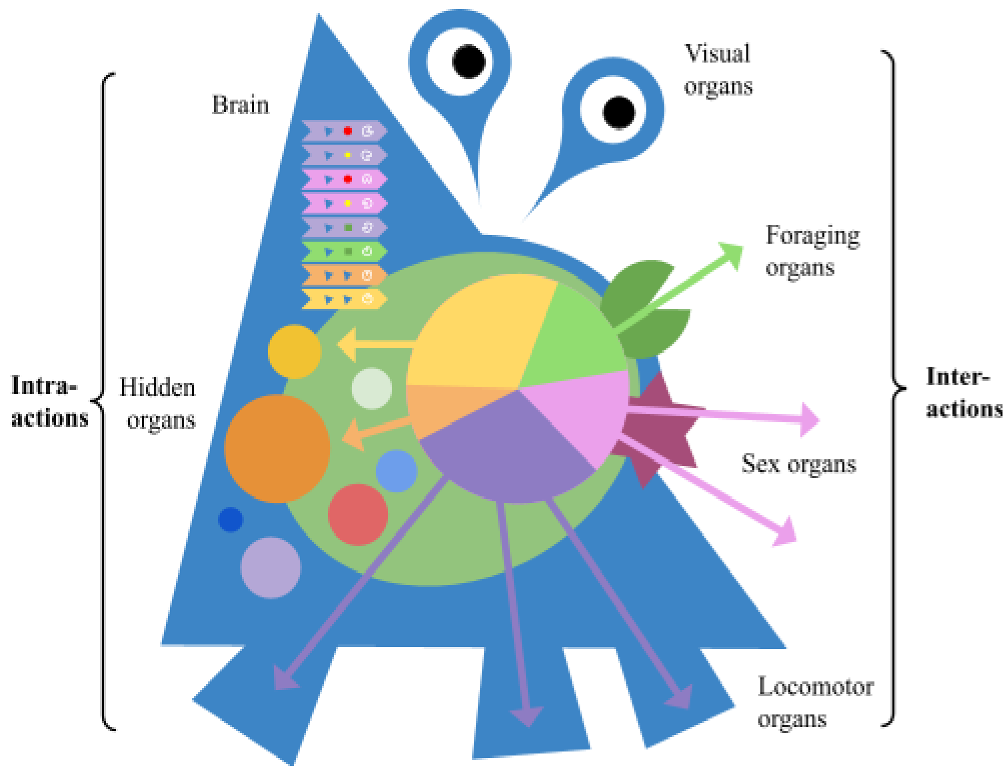

Figure 2 (b). A visual representation of the virtual **`animal`** male depicted in (a). The pie pieces represent the decided amount of energy for each decision and the arrows represent the actions being taken. Actions can only be taken if there is enough `energy.supply` to pay. During this process, the **`animal`** invests in its `survival.chance` and `conception.chance`, depicted as circles adjacent to other attributes.

Global operations. The B3GET global operations involve all subroutines that are directly accessed by **`animals`**. Essentially, this collection of subroutines represent the things that happen "for free," in that no **`animal`** pays energy for them to occur. This distinction is important in evolutionary models because complete knowledge of the energetics of a system is important to draw informative conclusions from the simulation output. In B3GET, these subroutines define the structure of the B3GET world, which together with the **`plant`** and **`animal`** dynamics described above, includes subroutines that impose deterioration, death, and decay on the animals. During this process, B3GET decreases each **`animal's`** `survival.chance` in proportion to the ***deterioration-rate***. Next, B3GET calculates each **`animal's`** survival for that timestep based on its `survival.chance`. Thus, in each simulation, **animals** must combat deterioration by investing in increasing their own `survival.chance` while simultaneously investing in reproductive strategies to make copies of their genes.

Another important operation that occurs globally is to update the territories for each group within the simulation. During this process, B3GET updates the plant on which each **animal** is currently standing to record the `group.identity` for that **animal**. Each **plant** keeps records of the last 100 **animals** to visit its cell in `pgroups.here` and sets its current group affiliation, `pgroup.identity`, as the mode group in its records. For virtual organisms who are territorial, this likely results in distinct territory boundaries indicated for these **plants**.