
blossom Documentation

Release 0.1

Bryan Brzycki

Dec 07, 2018

Contents

1 Table of Contents	3
2 Indices and tables	11
Python Module Index	13

blossom is a Python package for producing simulations of evolving organisms.

CHAPTER 1

Table of Contents

1.1 Installation

You can use pip to install the latest version of the package automatically:

```
pip install blossom
```

Alternately, execute:

```
git clone git@github.com:blossom-evolution/blossom.git
python setup.py install
```

1.2 Cookbook

1.3 Full Documentation

1.3.1 Subpackages

blossom.organism_behavior package

Submodules

blossom.organism_behavior.action module

`blossom.organism_behavior.action.move_and_drink(organism, organism_list, world)`

Move and drink. Each occurs with probability 1/2.

```
blossom.organism_behavior.action.move_and_reproduce(organism, organism_list, world)
```

Move and reproduce. Reproduction occurs with probability 1/8.

```
blossom.organism_behavior.action.move_only(organism, organism_list, world)
```

Only move.

```
blossom.organism_behavior.action.move_reproduce_drink(organism, organism_list, world)
```

Move, drink, and reproduce. Reproduction occurs with probability 1/8. Drinks with probability 3/8, and moves with probability 1/2.

blossom.organism_behavior.drinking module

```
blossom.organism_behavior.drinking.constant_drink(organism, organism_list, world)
```

Intake constant amount of water from world if water is present.

blossom.organism_behavior.movement module

```
blossom.organism_behavior.movement.simple_random(organism, organism_list, world)
```

Move in random direction with equal probability. For 2D, organisms walk diagonally.

```
blossom.organism_behavior.movement.stationary(organism, organism_list, world)
```

Organism stays still.

blossom.organism_behavior.reproduction module

```
blossom.organism_behavior.reproduction.pure_replication(organism, organism_list, world)
```

Replace organism with two organism with similar parameters. Essentially, only differences in parameters are organism id, ancestry, age, and water / food levels.

Module contents

1.3.2 Submodules

1.3.3 blossom.dataset_io module

Load information from a certain dataset, e.g. to resume a simulation, and write world and organism data back to file.

```
blossom.dataset_io.load_organism_dataset(fn)
```

Load dataset file from JSON. filenames can be a single string or a list of strings.

Parameters `fn` (`str`) – Input filename of saved organism dataset.

Returns `organism_list` – A list of Organism objects reconstructed from the saved dataset.

Return type list of Organisms

```
blossom.dataset_io.load_world_dataset(fn)
```

Load dataset file from JSON. filenames can be a single string or a list of strings.

Parameters **fn** (*str*) – Input filename of saved world dataset.

Returns **world** – World object reconstructed from the saved dataset.

Return type *World*

```
blossom.dataset_io.write_organism_dataset(organism_list, fn)
```

Write organism data from list of Organism objects to file in JSON format.

Parameters

- **organism_list** (*list of Organisms*) – List of Organisms to write to file.
- **fn** (*str*) – Output filename of saved organism dataset.

```
blossom.dataset_io.write_world_dataset(world, fn)
```

Write world information from World object to file in JSON format.

Parameters

- **world** (*World*) – World attributes to write to file.
- **fn** (*str*) – Output filename of saved world dataset.

1.3.4 blossom.fields module

Built-in dictionaries with world, species, and organism parameters (or fields).

Both Organism and World objects are initialized based on these field dictionaries, and values are either populated from parameter files or take on the default values specified in this module.

1.3.5 blossom.organism module

```
class blossom.organism.Organism(init_dict={})
```

Bases: object

A basic organism structure for all species.

act (*organism_list, world*)

Method that decides and calls an action for the current timestep. Searches through custom methods and built-in movement methods. The action method specifically selects an action to take, from “move”, “reproduce”, “drink”, and “eat”. Then the appropriate instance method from this class is executed to yield the final list of affect organisms.

Parameters

- **organism_list** (*list of Organisms*) – List of organisms, with which this organism may interact.
- **world** (*World*) – World, with which this organism may interact.

Returns **affected_organisms** – Organism or list of organisms affected by this organism’s action.

Return type list of Organisms

at_death (*cause*)

Check various conditions for death.

Parameters **cause** (*str*) – Potential cause of this organism’s death.

Returns **is_dead** – Returns True if organism is dead from the specified cause, False otherwise.

Return type bool

classmethod **clone** (*organism*)

Makes a new Organism object identical to the current one.

Parameters **organism** (*Organism*) – Organism to copy.

Returns **new_organism** – Copied organism.

Return type *Organism*

die (*cause*)

Method that “kills” organism.

Parameters **cause** (*str*) – Cause of this organism’s death.

Returns **dead_organism** – New “dead” state of this organism.

Return type *Organism*

drink (*organism_list*, *world*)

Method for handling drinking. Searches through custom methods and built-in drinking methods.

Parameters

- **organism_list** (*list of Organisms*) – List of organisms, with which this organism may interact.
- **world** (*World*) – World, with which this organism may interact.

Returns **affected_organisms** – Organism or list of organisms affected by this organism’s drinking.

Return type Organisms, or list of Organisms

eat (*organism_list*, *world*)

Method for handling eating. Searches through custom methods and built-in eating methods.

Parameters

- **organism_list** (*list of Organisms*) – List of organisms, with which this organism may interact.
- **world** (*World*) – World, with which this organism may interact.

Returns **affected_organisms** – Organism or list of organisms affected by this organism’s eating.

Return type Organisms, or list of Organisms

move (*organism_list*, *world*)

Method for handling movement. Searches through custom methods and built-in movement methods.

Parameters

- **organism_list** (*list of Organisms*) – List of organisms, with which this organism may interact.
- **world** (*World*) – World, with which this organism may interact.

Returns **affected_organisms** – Organism or list of organisms affected by this organism’s movement.

Return type Organisms, or list of Organisms

reproduce (*organism_list*, *world*)

Method for handling reproduction. Searches through custom methods and built-in reproduction methods.

Parameters

- **organism_list** (*list of Organisms*) – List of organisms, with which this organism may interact.
- **world** (*World*) – World, with which this organism may interact.

Returns **affected_organisms** – Organism or list of organisms affected by this organism’s reproduction. For example, this would include both parent and child organisms.

Return type Organisms, or list of Organisms

step (*organism_list*, *world*)

Steps through one time step for this organism. Reflects changes based on actions / behaviors and updates to health parameters.

Returns a list of organisms that the action produced (either new or altered organisms).

Parameters

- **organism_list** (*list of Organisms*) – List of organisms, with which this organism may interact.
- **world** (`World`) – World, with which this organism may interact.

Returns affected_organisms – Organism or list of organisms affected by this organism’s actions or health. This could be an updated version of this organism, especially if the organism dies during the time step, but could also be multiple other organisms affected by actions (i.e. children from reproduction).

Return type list of Organisms

update_age()

Increments age by 1.

update_food()

Updates health parameters relevant to food consumption.

Decreases current food level based on metabolism, and increments time without food accordingly. Note that organisms die of hunger if this reaches the maximum time without food.

update_parameter (*parameter, value, method='set'*)

Update a specific parameter of the organism.

Parameters

- **parameter** (*string*) – Parameter to update.
- **value** – Value with which to update.
- **method** (*string*) – Method types are: ‘set’, ‘add’, ‘subtract’.

Returns self – Same organism object with updated parameter.

Return type *Organism*

update_water()

Updates health parameters relevant to water consumption.

Decreases current water level based on metabolism, and increments time without water accordingly. Note that organisms die of thirst if this reaches the maximum time without water.

1.3.6 blossom.parameter_io module

Load information from parameter files and construct world and organism objects at the initial timestep.

```
blossom.parameter_io.load_species_parameters(fns,      init_world,      cus-  
                                              tom_methods_fns)
```

Load all available species parameter files.

Parameters

- **fns** (*list of str*) – Input filenames of species parameter files. Different species get different species parameter files, from which the individual organisms are initialized.
- **init_world** (`World`) – Initial World instance for this Universe.
- **custom_methods_fns** (*list of str*) – List of external Python scripts containing custom organism behaviors. `blossom` will search for methods within each filename included here.

Returns organism_list – A list of Organism objects constructed from the parameter file.

Return type list of Organisms

```
blossom.parameter_io.load_world_parameters(fn)  
Load world parameter file and construct initial World object.
```

Parameters **fn** (*str*) – Input filename of parameter file.

Returns **world** – World object constructed from the parameter file.

Return type *World*

1.3.7 blossom.parse_intent module

```
blossom.parse_intent.parse(intent_list, organism_list)
```

Determine whether the intent list is valid and fix it otherwise.

Parameters

- **intent_list** (*list of Organisms*) – List of organisms with proposed organism states, after each organism has ‘acted’
- **organism_list** (*list of Organisms*) – List of current organisms

Returns **updated_list** – List of updated organisms with conflicts between intent_list and organism_list resolved.

Return type list of Organisms

1.3.8 blossom.universe module

```
class blossom.universe.Universe(world_fn=None, organisms_fn=None, world_param_fn=None, species_param_fns=None, custom_methods_fns=None, current_time=0, end_time=10, dataset_dir='datasets/', pad_zeroes=4, file_extension='.txt')
```

Bases: *object*

Create the universe of the simulation.

initialize_organisms()

Initialize all organisms in the universe from either a saved dataset or from parameter files (and subsequently writing the initial time step to file).

Returns **organism_list** – List of organisms at the beginning of the simulation.

Return type list of Organisms

initialize_world()

Initialize the world of the universe from either a saved dataset or from a parameter file (and subsequently writing the initial time step to file).

Returns **world** – World at the beginning of the simulation.

Return type *World*

step()

Steps through one time step, iterating over all organisms and computing new organism states. Saves all organisms and the world to file at the end of each step.

1.3.9 blossom.utils module

Common utilities used throughout *blossom*

`blossom.utils.cast_to_list(x)`

Make a list out of the input if the input isn't a list.

1.3.10 blossom.world module

`class blossom.world.World(init_dict={})`

Bases: object

World class for the environment of the simulation.

1.3.11 blossom.world_generator module

`blossom.world_generator.constant_list(val, length)`

Generate a constant-valued list.

`blossom.world_generator.write_environment(water, food, obstacles, environment_fn='environment.json')`

Write water, food, and obstacles lists to an environment file.

1.3.12 Module contents

`blossom` is a package for simulating evolution

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

b

```
blossom, 9
blossom.dataset_io, 4
blossom.fields, 5
blossom.organism, 5
blossom.organism_behavior, 4
blossom.organism_behavior.action, 3
blossom.organism_behavior.drinking, 4
blossom.organism_behavior.movement, 4
blossom.organism_behavior.reproduction,
    4
blossom.parameter_io, 7
blossom.parse_intent, 8
blossom.universe, 8
blossom.utils, 9
blossom.world, 9
blossom.world_generator, 9
```

Index

A

act() (blossom.organism.Organism method), 5
at_death() (blossom.organism.Organism method), 5

B

blossom (module), 9
blossom.dataset_io (module), 4
blossom.fields (module), 5
blossom.organism (module), 5
blossom.organism_behavior (module), 4
blossom.organism_behavior.action (module), 3
blossom.organism_behavior.drinking (module), 4
blossom.organism_behavior.movement (module), 4
blossom.organism_behavior.reproduction (module), 4
blossom.parameter_io (module), 7
blossom.parse_intent (module), 8
blossom.universe (module), 8
blossom.utils (module), 9
blossom.world (module), 9
blossom.world_generator (module), 9

C

cast_to_list() (in module blossom.utils), 9
clone() (blossom.organism.Organism class method), 6
constant_drink() (in module blossom.organism_behavior.drinking), 4
constant_list() (in module blossom.world_generator), 9

D

die() (blossom.organism.Organism method), 6
drink() (blossom.organism.Organism method), 6

E

eat() (blossom.organism.Organism method), 6

I

initialize_organisms() (blossom.universe.Universe method), 8
initialize_world() (blossom.universe.Universe method), 8

L

load_organism_dataset() (in module blossom.dataset_io), 4
load_species_parameters() (in module blossom.parameter_io), 7
load_world_dataset() (in module blossom.dataset_io), 4
load_world_parameters() (in module blossom.parameter_io), 8

M

move() (blossom.organism.Organism method), 6
move_and_drink() (in module blossom.organism_behavior.action), 3
move_and_reproduce() (in module blossom.organism_behavior.action), 3
move_only() (in module blossom.organism_behavior.action), 4
move_reproduce_drink() (in module blossom.organism_behavior.action), 4

O

Organism (class in blossom.organism), 5

P

parse() (in module blossom.parse_intent), 8
pure_replication() (in module blossom.organism_behavior.reproduction), 4

R

reproduce() (blossom.organism.Organism method), 6

S

simple_random() (in module blossom.organism_behavior.movement), 4
stationary() (in module blossom.organism_behavior.movement), 4
step() (blossom.organism.Organism method), 6
step() (blossom.universe.Universe method), 8

U

Universe (class in blossom.universe), [8](#)
update_age() (blossom.organism.Organism method), [7](#)
update_food() (blossom.organism.Organism method), [7](#)
update_parameter() (blossom.organism.Organism
method), [7](#)
update_water() (blossom.organism.Organism method), [7](#)

W

World (class in blossom.world), [9](#)
write_environment() (in module blossom.
som.world_generator), [9](#)
write_organism_dataset() (in module blossom.
som.dataset_io), [5](#)
write_world_dataset() (in module blossom.dataset_io), [5](#)