SLiM

Workshop Series

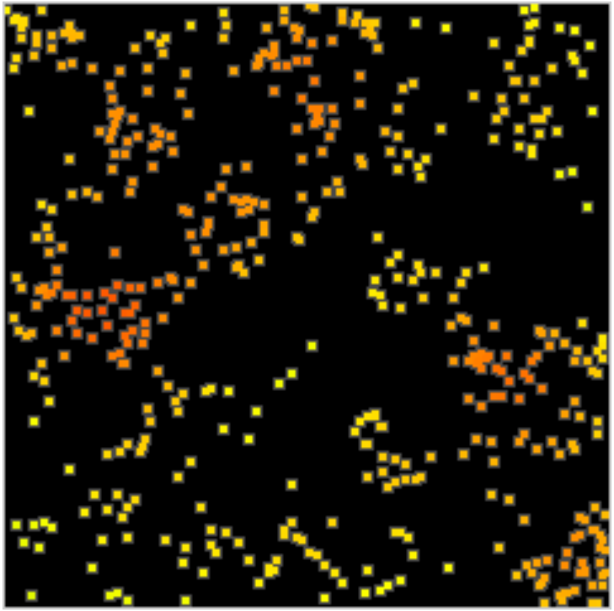
#18: Continuous Space I

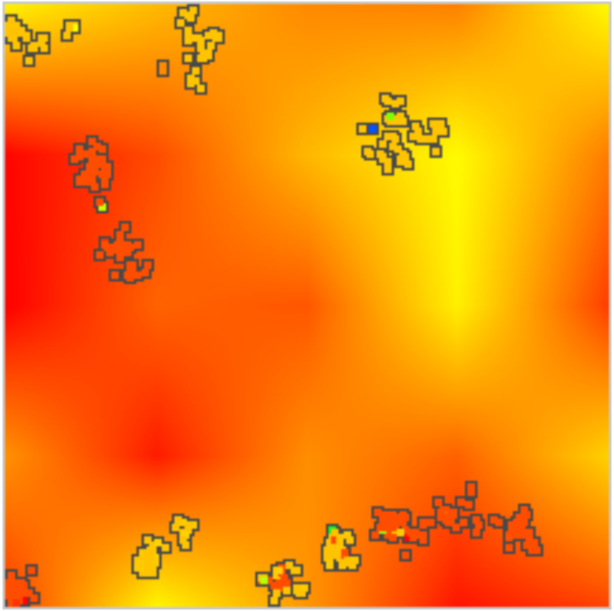
Benjamin C. Haller

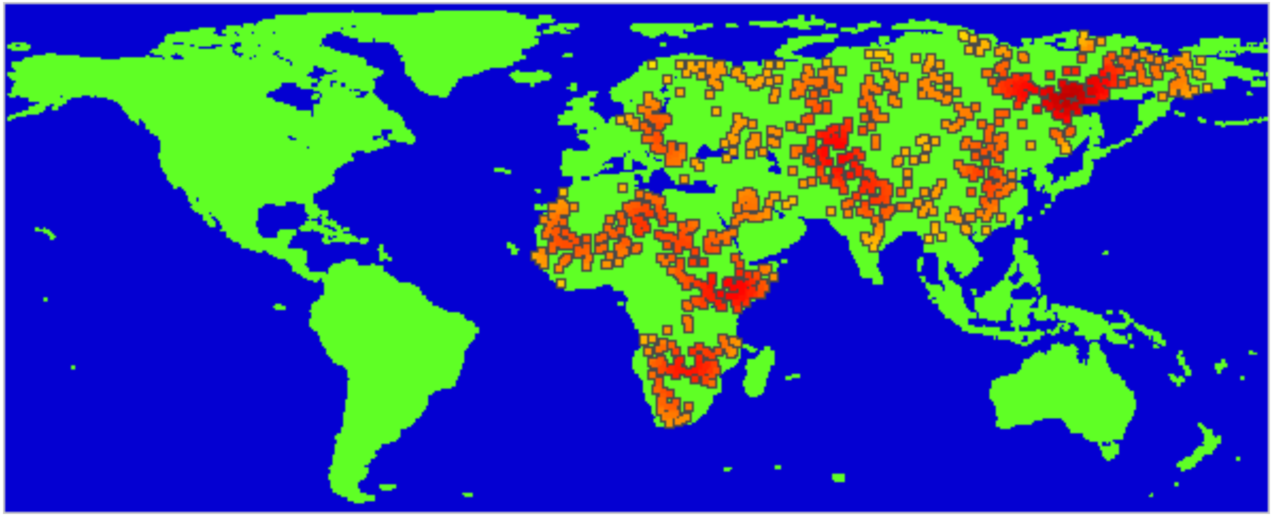
Messer Lab, Cornell University

Continuous Space

- Up to now: discrete subpopulations
- Continuous space entails:
  - a continuous 1D, 2D, or 3D landscape
  - boundary conditions at the edges
  - spatial coordinates for individuals
  - spatial interactions with `InteractionType`
  - spatial maps: terrain, habitability, resources







Continuous Space

- Declare the model as spatial
  - `initializeSLiMOptions(dimensionality="xy")`
- Set individual positions
  - `x`, `y`, `z`, `spatialPosition`, `setSpatialPosition()`
- Define spatial interactions
  - `initializeInteractionType()`
  - `setInteractionFunction()`
- Define spatial maps
  - `defineSpatialMap()`

The `initialize()` callback

```
initialize() {
    initializeSLiMOptions(dimensionality="xy");
    initializeMutationRate(1e-7);
    initializeMutationType("m1", 0.5, "f", 0.0);
    initializeGenomicElementType("g1", m1, 1.0);
    initializeGenomicElement(g1, 0, 99999);
    initializeRecombinationRate(1e-8);
}
```

- Calls `initializeSLiMOptions()`
- Defines the **dimensionality**:
  - `"x"`
  - `"xy"`
  - `"xyz"`

The `1 late()` event

```
1 late() {
    sim.addSubpop("p1", 500);

    // initial positions are random in ([0,1], [0,1])
    p1.individuals.x = runif(p1.individualCount);
    p1.individuals.y = runif(p1.individualCount);
}
```

- Sets up initial spatial positions: `x` and `y`
- Vectorized assignment from `runif()`
- We will see better ways:
  - `setSpatialPosition()`
  - `pointUniform()`

The `modifyChild()` callback

```
modifyChild() {
    // draw a child position near the first parent
    do
        child.x = parent1.x + rnorm(1, 0, 0.02);
    while ((child.x < 0.0) | (child.x > 1.0));

    do
        child.y = parent1.y + rnorm(1, 0, 0.02);
    while ((child.y < 0.0) | (child.y > 1.0));

    return T;
}
```

- Sets the child's `x` and `y`
- We will see better ways:
    - `setSpatialPosition()`, `pointInBounds()`

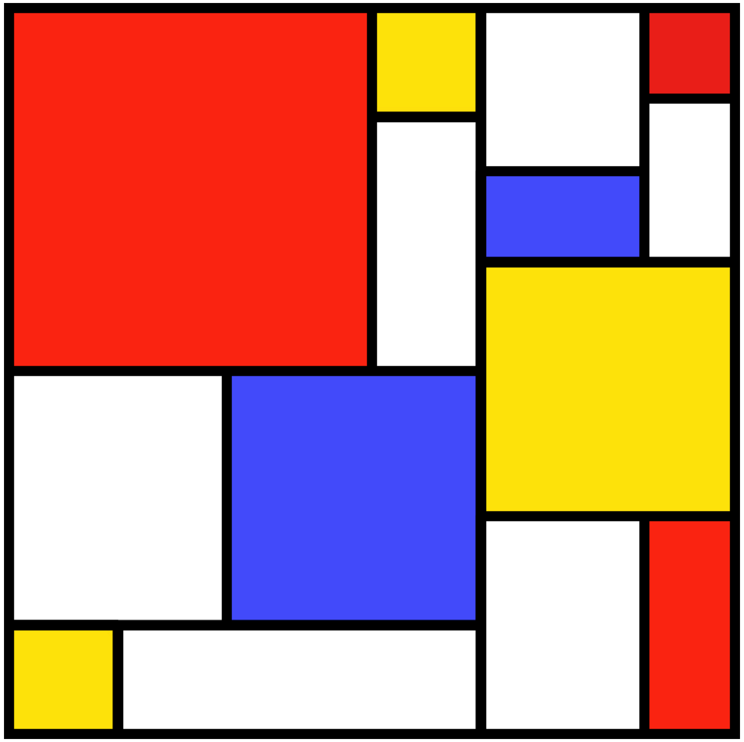
A complete spatial model

```
initialize() {
    initializeSLiMOptions(dimensionality="xy");
    initializeMutationRate(1e-7);
    initializeMutationType("m1", 0.5, "f", 0.0);
    initializeGenomicElementType("g1", m1, 1.0);
    initializeGenomicElement(g1, 0, 99999);
    initializeRecombinationRate(1e-8);
}
1 late() {
    sim.addSubpop("p1", 500);

    // initial positions are random in ([0,1], [0,1])
    p1.individuals.x = runif(p1.individualCount);
    p1.individuals.y = runif(p1.individualCount);
}
modifyChild() {
    // draw a child position near the first parent
    do
        child.x = parent1.x + rnorm(1, 0, 0.02);
    while ((child.x < 0.0) | (child.x > 1.0));

    do
        child.y = parent1.y + rnorm(1, 0, 0.02);
    while ((child.y < 0.0) | (child.y > 1.0));

    return T;
}
2000 late() { sim.outputFixedMutations(); }
```



SLiM Workshop Exercise #18

Copyright © 2020–2021 Benjamin C. Haller. All Rights Reserved.