

## **ROOT DISEASE MODEL (v.1.5.1)**

### **WHAT IS IT?**

This is a model of root disease spread between trees in the landscape. The disease spreads via two transmission processes: (a) root contact/root graft transmission between adjacent trees and (b) insect vectors that carry spores between trees.

### **HOW IT WORKS**

This model has a large number of variations designed to test the nature of the system under different conditions and thus explore hypotheses.

### **Agents**

#### ***Patches***

Each patch represent the space for one tree. The identity of that patch is indicated by the color and properties, including tree (true/false), alive (true/false), infected (true/false), and “inf-source” (the cause of the infection, either “initial” from the SETUP, “vector”, “root” from root spread, or “NA” for uninfected trees).

This disease is seeded into the landscape at a user-defined proportion of all patches and then spreads between adjacent trees via root systems and when carried by insect vectors.

Each patch represents a tree or an empty space where a tree could be. Each tree is either alive or dead, whole or a stump, infected or not infected. Trees also have an age, which is updated at each tick. Finally, trees have an “infection countdown” property, discussed later.

#### ***Patch colors:***

- **Live, uninfected tree:** Green
- **Live, infected tree:** Yellow
- **Dead, uninfected tree/stump:** Brown
- **Dead, infected tree/stump:** Dark red/maroon
- **Empty space:** White

#### ***Turtles***

Each turtle (bug-shaped agent) represents an insect vector. The vector has only one property: carrier (true/false), which indicates if the insect carries the disease or not and thus whether it can infect new trees. The number of turtles remains constant throughout the model, and no new turtles are born or die.

#### ***Turtle colors:***

- **Non-carrier:** Black
- **Carrier:** Red

### **Model procedures**

#### **Startup:**

Upon opening the model, all user-defined parameters are set to default values as determined by the “set-initial-config” procedure. If parameter values are changed by the user, the “Default configuration” button can be used to reset parameters to these default values.

**Setup:**

The model is cleared, the time set represented by each tick is set, and the properties of all patches are set to empty space values.

Patches are set to create live trees, either placed in a checker-board pattern if the “checker?” switch is set to “on” (always the same number of trees) or randomly at the proportion indicated by the “tree-density” slider. A. All trees are set to the same age, as determined by the “initial-stand-age” slider. B. Some proportion of the trees are set to infected, based on the “infection-density” slider.

A number of insect vectors (turtles) are created according to the “num-vectors”. Some proportion of the vectors are set as carriers as determined by the “pct-initial-carriers” slider if the “carriers?” switch is set to “on”.

Default colors are set and the “world” is colored.

Counts are set as globals and (some) are displayed in the monitors (total # of trees, # of live trees, # of dead trees, # of uninfected trees, # of infected trees, # of empty patches).

Tick count is reset to 0.

**Go (iterative/tick):**

1. STOP PROCEDURE: If there are no more live, uninfected trees, the model stops and returns the message, “Black stain has inherited the earth.” Similarly, if there are no more infected trees and no disease-carrying vectors, the model stops and returns the message, “Black stain has been eradicated!” If there are no live trees and the “replant?” switch is off, the model stops and returns the message, “All the trees are gone!”
2. AGING: Each tree’s age increases by the value “time-step”.
3. CHECK FUNGAL VIABILITY: Fungal viability is checked in dead trees. Once an infected tree dies, the “inf-countdown” property is set to the value of the “fungal-viability” slider. The dead tree will remain infected for as long as the value of “inf-countdown” is greater than 0. If it is less than or equal to 0, the dead, infected tree will become uninfected. This reflects the fact that, in the real world, fungi in dead trees loses the ability to infect new trees over time.
4. REPLANTING: Replanting occurs (dead trees set to alive and age reset) if “replant?” switch is set to “on”. All dead trees are set to alive and their age is set to 0. Roughly half of dead, infected trees will be replanted as already infected because the root disease pathogen can linger alive in the soil. Trees are always replanted in the same spot in this model version, with empty cells forever remaining empty.
5. THINNING: Thinning occurs if the “thinning?” switch is on. Some percentage of trees (“pct-thin” slider) at the user defined thinning age (“thin-age” slider) will be killed and have their age set to 0. For dead, infected trees, the fungal viability countdown begins.
6. HARVEST: Harvest occurs if the “harvest?” switch is on. All trees (“pct-thin” slider) at the user defined harvest age (“harvest-age” slider) will be killed and have their age set to 0. For dead, infected trees, the fungal viability countdown begins.

7. VECTOR INFECTION: - Trees are infected via vectors. This consists of three steps outlined in the “vector-infection” procedure:
  - a. Vectors move: i. If the “attraction?” switch is ON, an attraction cone is created for each vector, consisting of a set of infected patches: 1 and 2 ahead, left 45 degrees and 1 and 2 ahead, left 90 degrees and 1 and 2 ahead, right 45 degrees and 1 and 2 ahead, and right 90 degrees and 1 and 2 ahead. If there are infected patches in this attraction-cone, the vector moves to one of those patches. If there are no infected patches in this attraction cone, the vector turns to a random direction (0 to 360 degrees) and moves forward as determined by the “dispersal-distance” slider. ii. If the “attraction?” switch is OFF, the vector turns to a random direction (0 to 360 degrees) and moves forward as determined by the “dispersal-distance” slider.
  - b. Infect: Vectors can infect trees if the vector is a carrier. The probability of this occurring is based on the “prob-vector-infection” slider.
  - c. Infest: Vectors can become carriers if they are on a patch that is an infected tree. The probability of this occurring is based on the “prob-spore-attach” slider.
8. ROOT INFECTION: - Live, infected trees can infect one of the patches at a certain distance from them (based on “root-infection-radius” slider). The probability that this occurs is based on the “prob-root-spread” slider. This represents the real-world process of fungal growth between adjacent trees with very close or connected root systems.
9. COLOR WORLD: The colors of the patches and turtles are updated to reflect their properties.
10. SET GLOBALS: The globals, including counts of patches with particular characteristics, are updated to reflect the current state of the world.

## HOW TO USE IT

Choose initial parameter settings using the sliders and switches or select the default configuration using the button.

### INPUTS:

GENERAL stop? - Whether or not the model will stop on its own given certain states of the world. If OFF, the model will run forever regardless of the state of the world.

WORLD checker? - When ON, sets the initial forest to a checkerboard pattern (ignores “tree-density” slider) tree-density - The proportion of patches that are trees (rather than empty spaces) infection-density - The initial proportion of trees that are infected initial-stand-age - The initial age of all trees in the world num-vectors - Sets the number of vectors (turtles) present. This does not change throughout the model run. pct-initial-carriers - Sets the number of vectors that are carriers after running the SETUP.

VECTORS attraction? - Determines whether vectors move to infected trees in their immediate vicinity rather than following the normal move procedure. dispersal-distance - The distance that each vector moves at each tick. The only exception to this is when the “attraction?” switch is ON. prob-spore-attach - The probability that a non-carrier vector will become a carrier when it is at an infected tree

patch. prob-vector-infection - The probability that a non-infected tree will become infected when a carrier vector is present.

TREES root-infection-radius - The maximum distance that an infected tree can infect another via root spreading. prob-root-spread - The probability that the infection will spread via the “root-infection” procedure for each patch at every time step.

MANAGEMENT replant? - Whether dead trees are replanted after a harvest. thinning? - Whether or not the thinning procedure will run. thin-age - The age at which the thinning procedure will be applied, if “thinning?” is ON. pct-thin - The proportion of live trees that will be killed when the thinning procedure runs. harvest? - Whether or not the harvest procedure will run. harvest-age - The age at which trees will be harvested (killed).

Click SETUP to create the forest and the vectors.

Click GO ONCE (one tick) or GO FOREVER to start the simulation.

Note that some parameter settings will not take effect until the next SETUP.

Outputs of particular interest given initial conditions include: \* Percent of trees infected over time \* Relative amount of infection from root vs. vector infection (see infection source plot) \* Amount of carriers \* Time to full infection \* Time to disease eradication

## **THINGS TO NOTICE**

What conditions lead to root infection being the dominant mechanism of disease spread? What conditions lead to vectors being the dominant mechanism of disease spread?

What management conditions (e.g. thinning, harvest, age of management actions, percent of forest thinned) lead to disease eradication versus the entire forest becoming infected?

## **THINGS TO TRY**

Play around!

## **EXTENDING THE MODEL**

Setting maximum dispersal, stochastic dispersal

Dead trees can become infected for some limited amount of time

Changing number of vectors (generations)

## **NETLOGO FEATURES**

Uses “neighbors”, “in-radius”, and other unique commands.

## **RELATED MODELS**

Fire, Spread of Disease

## **CREDITS AND REFERENCES**

Adam Bouché is the sole author of this model. References for code were taken from experience playing with other models in the NetLogo library and in the Intro to Agent-based Modeling textbook by Wilensky and Rand.

## **COPYRIGHT AND LICENSE**

Though no formal copyright or license yet exists, Adam Bouché (the author) intends to pursue a Creative Commons License.

## **ACKNOWLEDGEMENTS**

Thank you to Dr. Bill Rand for teaching this course on NetLogo and Agent-Based Modeling and Dr. Uri Wilensky for creating and contributing to NetLogo.

Thank you to my professors for helping in the development of this project and the conceptual model on which this model is based.