**Notes**

## Modelling and Analyzing the Growth of a Network

### What is it?

This program models and analyzes the growth of a network based on the biological concept of cell growth and structure formation. In this model, a cell with a nucleus reproduces by splitting off daughter cells, which are formed at defined positions (lower left and lower right of the parent cell). After this, the daughter cells repeat the cycle of separation and positioning. However, daughter cells that end up at the same location experience competition for space and resources and as a result become infertile. The outcome of the process is a tree-shaped "tissue" of green and red patches (respectively reproducing and non-reproducing cells).

### How it works

This program starts by creating a cell in the center of the top row of the canvas. The cell is colored green, indicating it is a reproducing cell. The program then proceeds to create daughter cells at the defined positions (lower left and lower right) of the parent cell, which are also colored green. These daughter cells also repeat the process, forming new cells at defined positions, which are also colored green. However, if two daughter cells end up at the same position, they compete for space and resources and become infertile. Infertile cells are colored red, indicating they are not reproducing cells.

### How to use it

To use this program:

Click the "Set-up" button to set up the initial agent.
Click the "go-once" button to run the program incrementally (+1).
Click the "go" button to run the program continuously.
Move the "#generations" slider to specify the number of generations outputted.
The "show-patch-color" switch clears the patches by setting their color to black.
The "show-circle?" switch arranges the nodes of the network as a circle.
Note: The program stops after it has been checked if generationstep = #generations.

### Things to notice

The growth of the network over time.
The change in color of each cell indicating whether it is a reproducing cell (green) or a non-reproducing cell (red).

## Things to try

Experiment with changing the number of generations and analyze the output using the tables provided.
Turn the "show-circle?" switch on and off to see the difference in the arrangement of nodes in the network.

## Credits

This script was created by Anthony Constant (AC). If you have any questions or suggestions, you can contact him at anthonyconstant.co.uk/

## License



## GitHub

Share Link: https://github.com/Anthony-Constant/Tissue-Growth

**NETLOGO COPY & PASTED LOCAL SOURCE CODE**

```
 # Modelling and Analyzing the Growth of a Network
# Created Tissue Growth Simulation in NetLogo
# Author: Anthony Constant (AC)

breed [children child]
breed [copies copy]
globals [generationstep AMPL CC]
copies-own [state]
extensions [nw]


to setup
 ca
 set show-patch-color TRUE
 set generationstep 1

 crt 1
  [
   setxy (max-pxcor / 2) (max-pycor)
   mark-and-breed
  ]

  reset-ticks
end

to mark-and-breed
  ifelse (pcolor != green)
    [set pcolor green replicate]
    [set pcolor red]
  die
end

to replicate
```

```
  hatch-children 1
     [
      set size 0.75
      set color orange
      set shape "triangle"
      if pcolor =  green [make-copy make-links]
     ]
end

to make-copy
 hatch-copies 2
   [
    set size 0.25
    set shape "circle"
    set color white
    if who mod 2 = 0 [set state  "L"]
   ]
end

to go
  set generationstep generationstep + 1

  ask copies
    [
      ifelse pcolor != red
       [move mark-and-breed][die]
    ]

  if generationstep = #generations
   [
    ask copies [die]
        if show-circle?
  [
    make-circle
  ]
  if show-patch-color = false
  [
    make-patch-color
```

```
    stop
   ]

  ]


  tick

  compute-metrics

end

to move
   ifelse state = "L"
    [setxy (xcor - 1) (ycor - 1)]
    [setxy (xcor + 1) (ycor - 1)]
end


to make-circle
  if show-patch-color = False
  [
  layout-circle turtles (world-width / 2 - 20)
  ]
end

to make-patch-color
  if show-patch-color = false
  [
clear-patches
  ]
end



to make-links
  ask children with [pcolor = green][create-links-with children-on neighbors with [pcolor = green]
  let child2 patch (xcor - 1) (ycor - 1) ask children-at  1 -1 [create-links-with children-on child2 ]]
```

```
  ask links [set color white]
end

to compute-metrics
  nw:set-context children links

  set AMPL nw:mean-path-length
  set CC mean [ nw:clustering-coefficient ] of children
end
```