Networks – Structure and Dynamics

Nathaniel Osgood

Using Modeling to Prepare for Changing Healthcare Needs

Duke-NUS

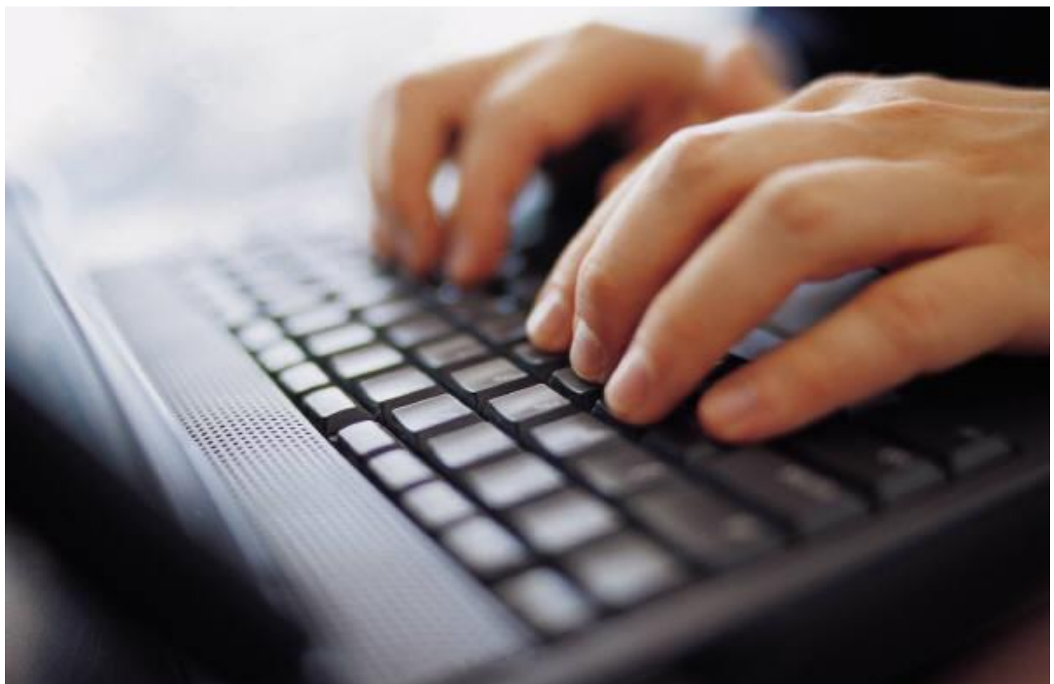
April 16, 2014

Announcements

- This Week's Extra Office Hours
  - Wednesday 3-5pm
  - Friday 2:30-4pm

- Take-Home Exercise for Thursday
  - Please come to class having tried this

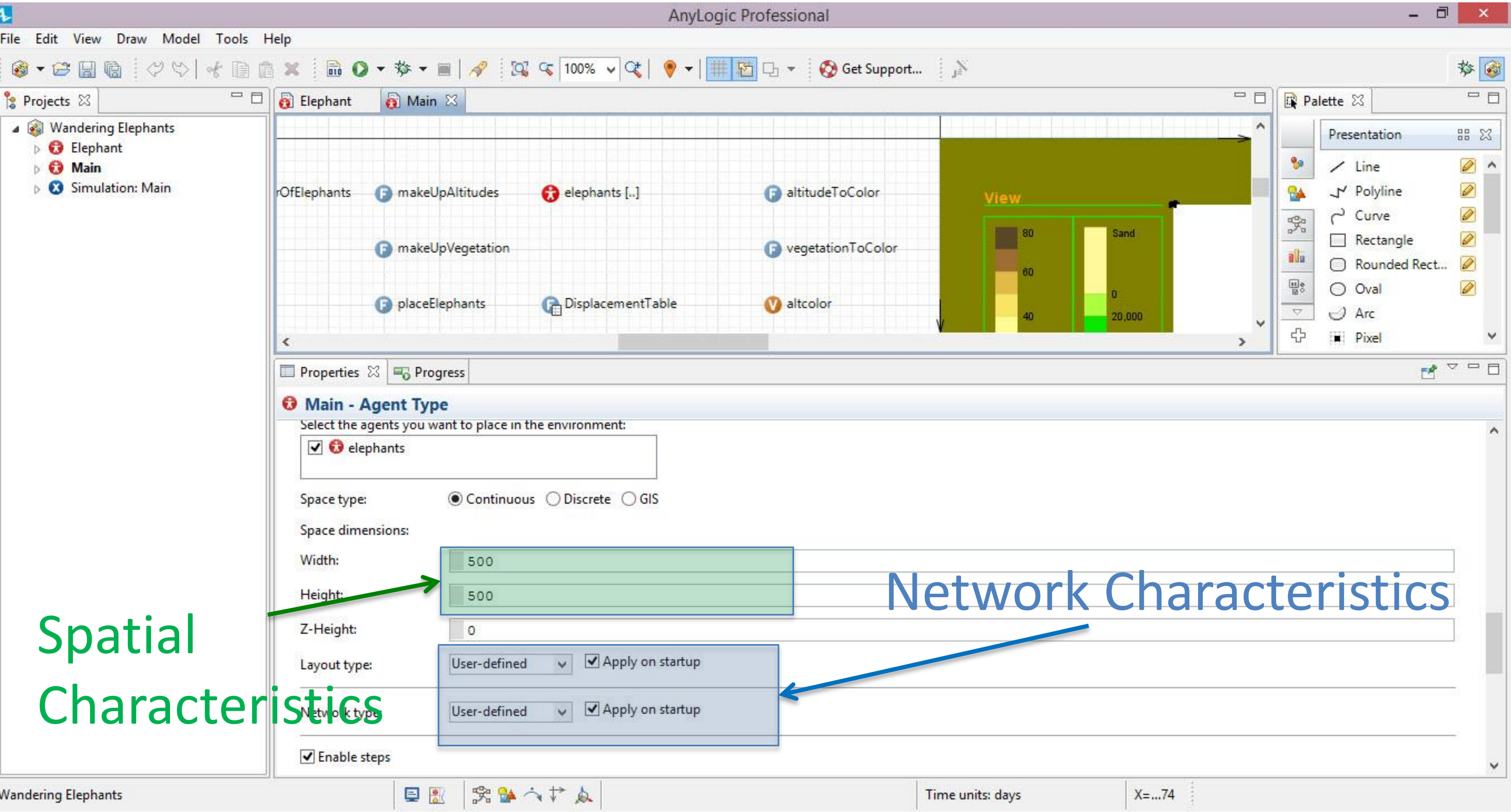
Hands on Model Use Ahead



Load your recently created SI model

(provided alternative: MinimalistSIRNetworkABM)

The *Environment* defines both Spatial & Network (Topological) Context



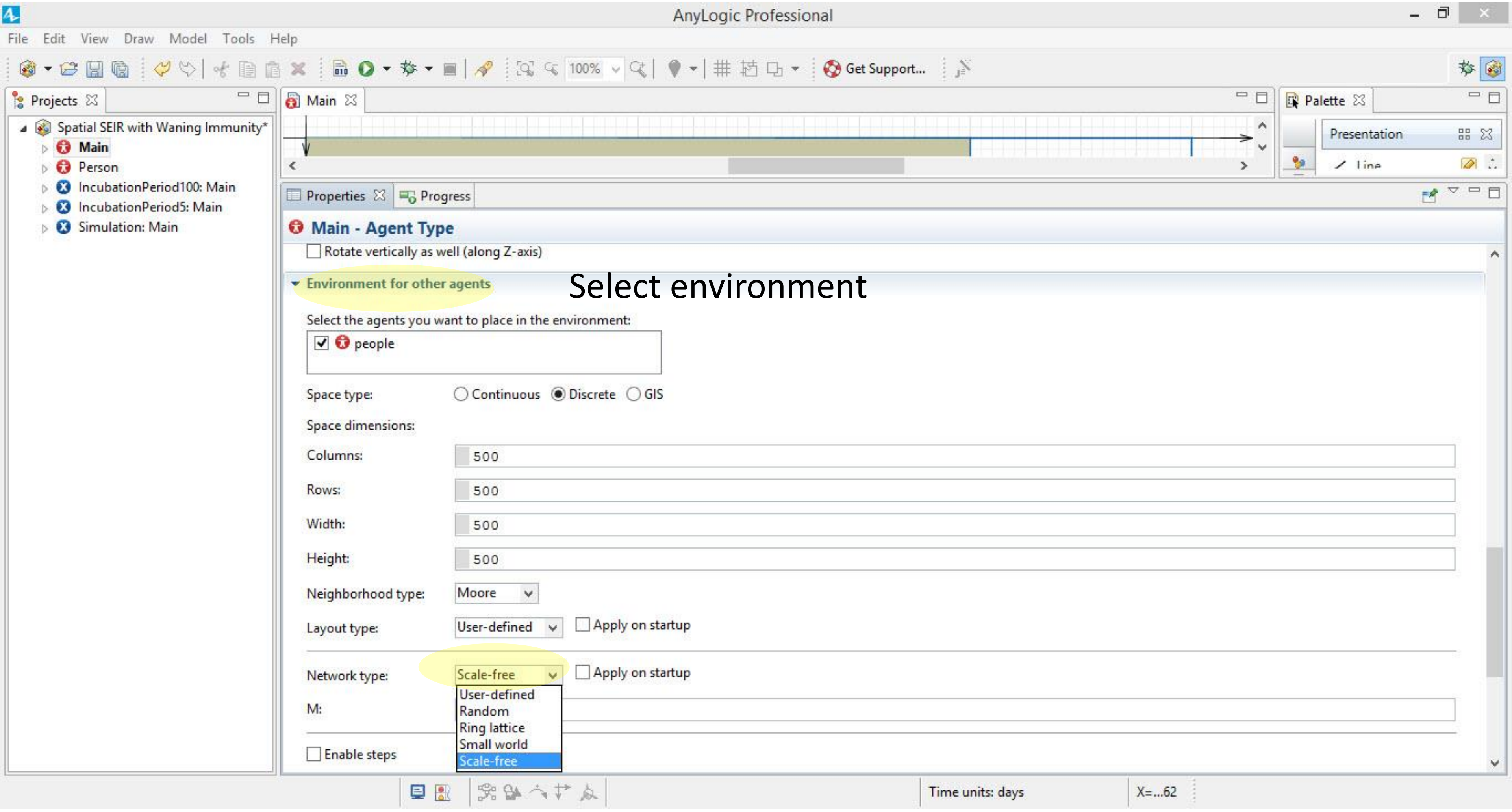
Network Specification in AnyLogic

- When considering networks in AnyLogic, we specify two somewhat distinct (but coupled) things
  - Network topologies
  - Spatial (and visual) Layouts

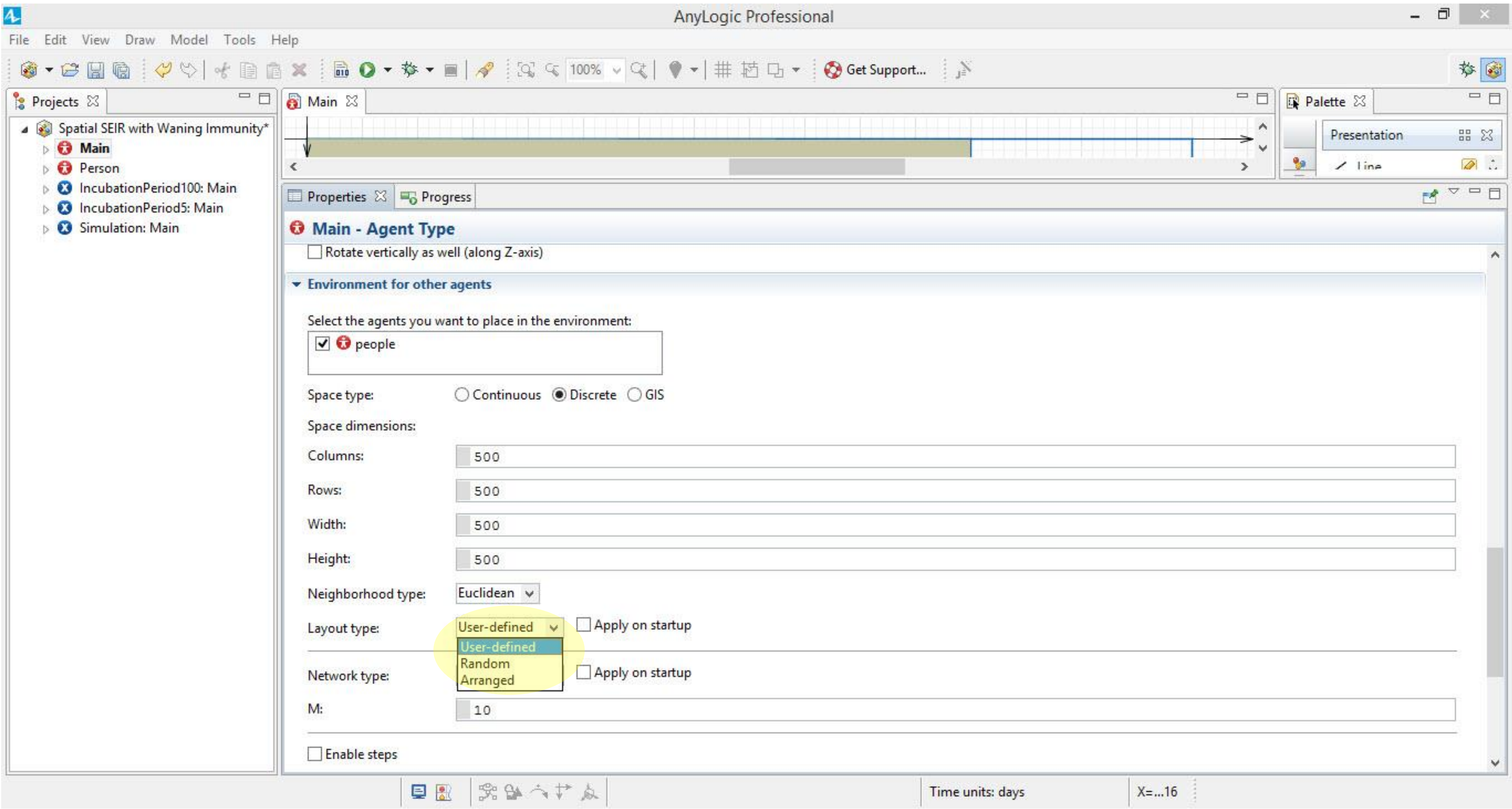
Networks & Spatial Layouts

- Distinct node attributes: Location & connections
  - Spatial layouts determine where nodes appear in space (and on the screen!)
  - Network type determines who is connected to whom
  - For the most part, these characteristics are determined independently
- Network topologies (connectedness) can be defined either *alternative to* or *in addition to spatial layouts*
  - Agents will have spatial locations in either case

Network Types



Layout Types



Layout Type

- **Random**: Uniformly distribute X and Y position of nodes
- **Arranged**: Set node locations in a regular fashion (normally in a 2D grid)
- **Ring**: Set node locations in periodically spaced intervals around a ring shape
- **Spring Mass**: Adjust node locations such that node locations that are most tightly connected tend to be closer together
  - (Sets location based on network!)
- **User-Defined** User can set location (e.g. in initialization code)

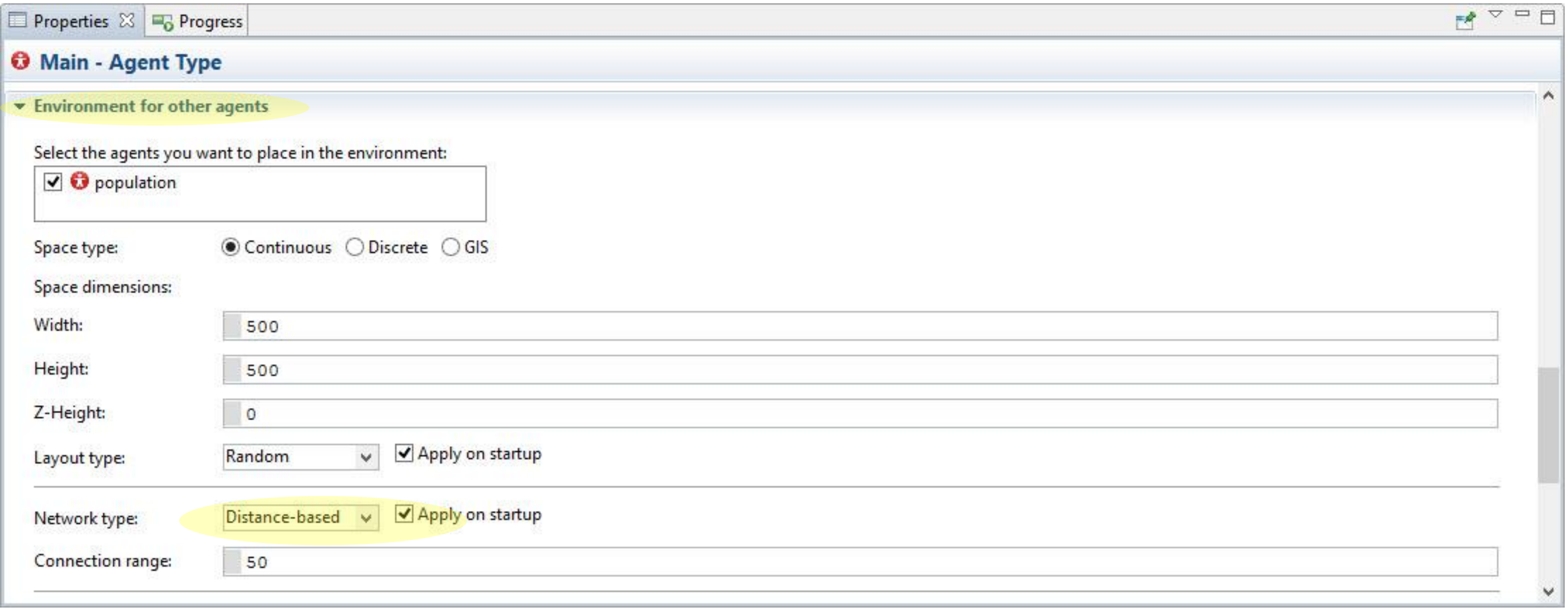
Distance Based Networks

- Function: Capturing geographic locality in networks
- Networks may be *discontinuous* (divided into disjoint *components*) when
  - The threshold is small
  - The density of the spaces (nodes per unit area) is too small

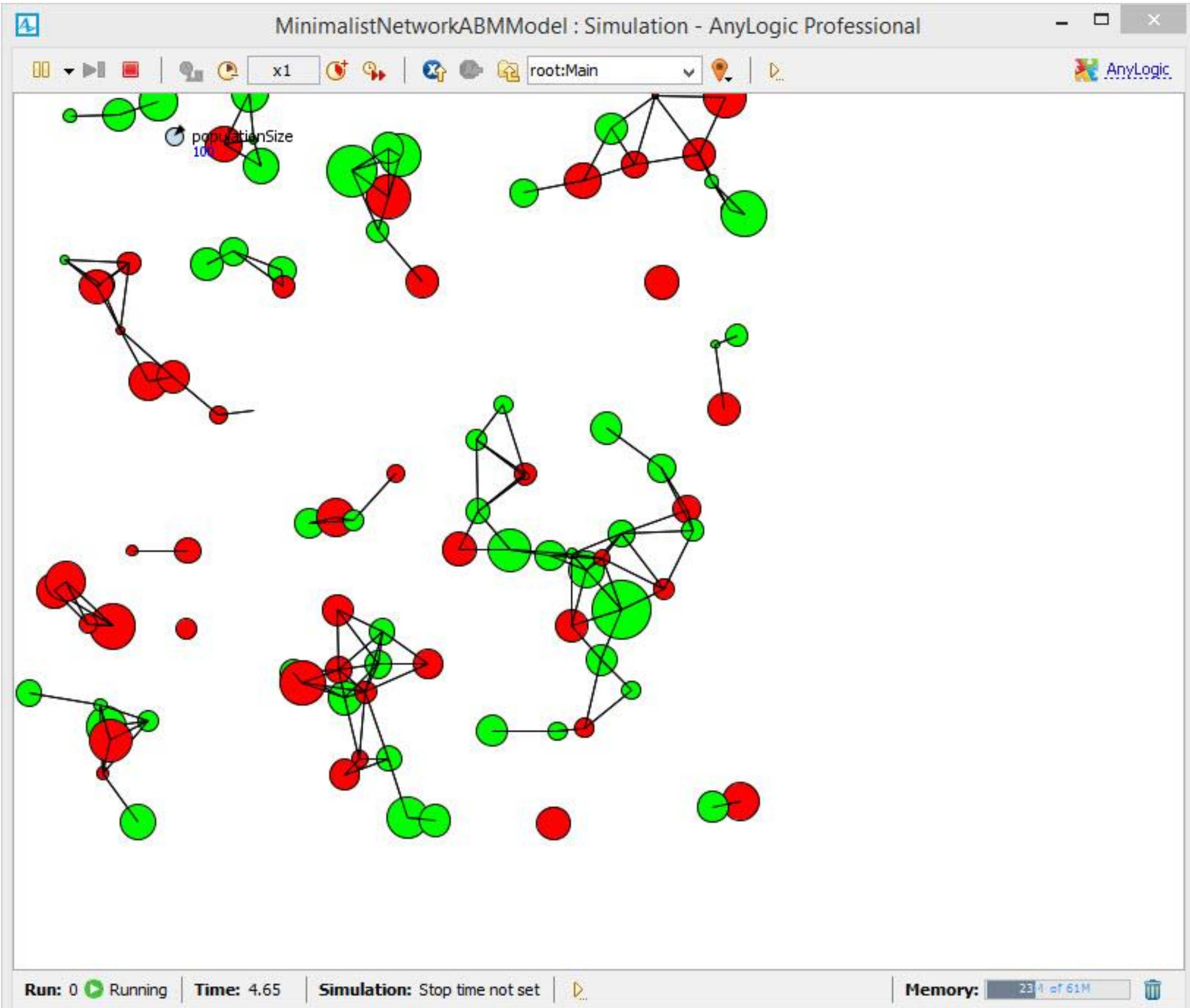
Interaction Between Network&Location 1

- For one type of networks (Distanced Based), whether there is a connection between A and B depends on the *distance* between A & B
  - This sets connectivity based on location considerations!

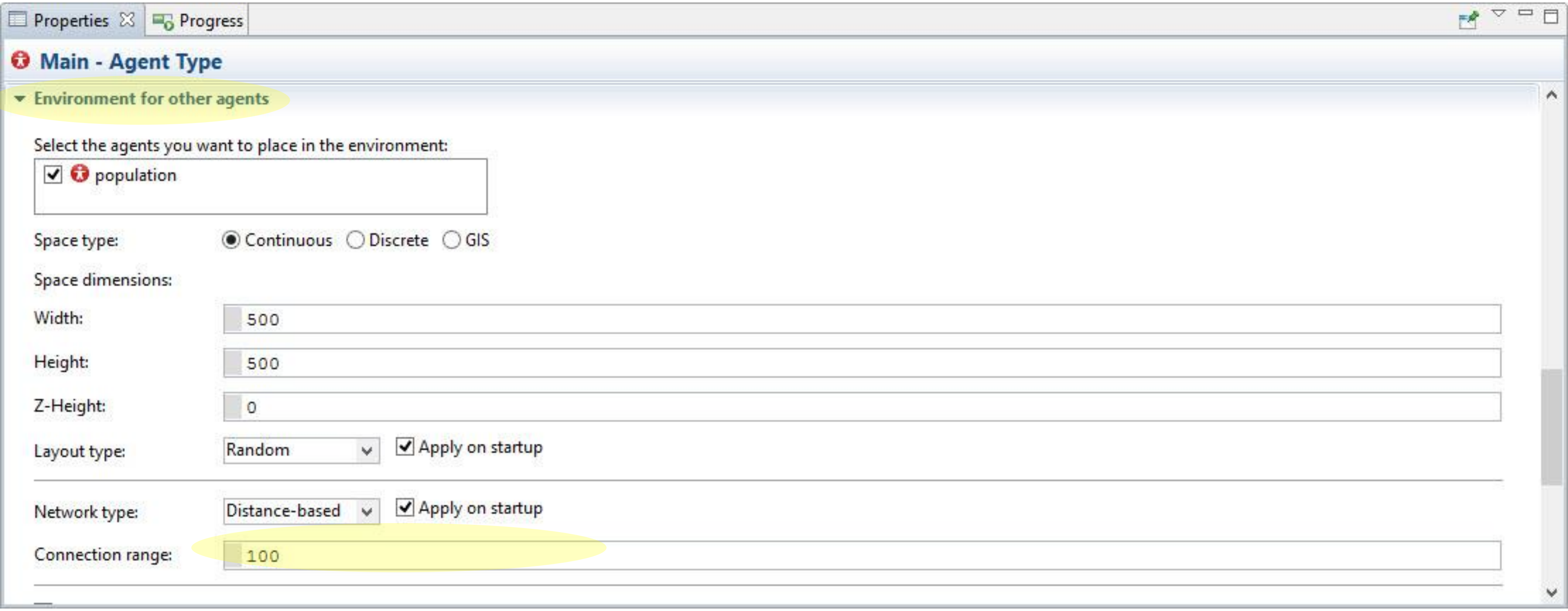
Property for Distance-Based Layout: Distance Threshold



Distance-Based Layout



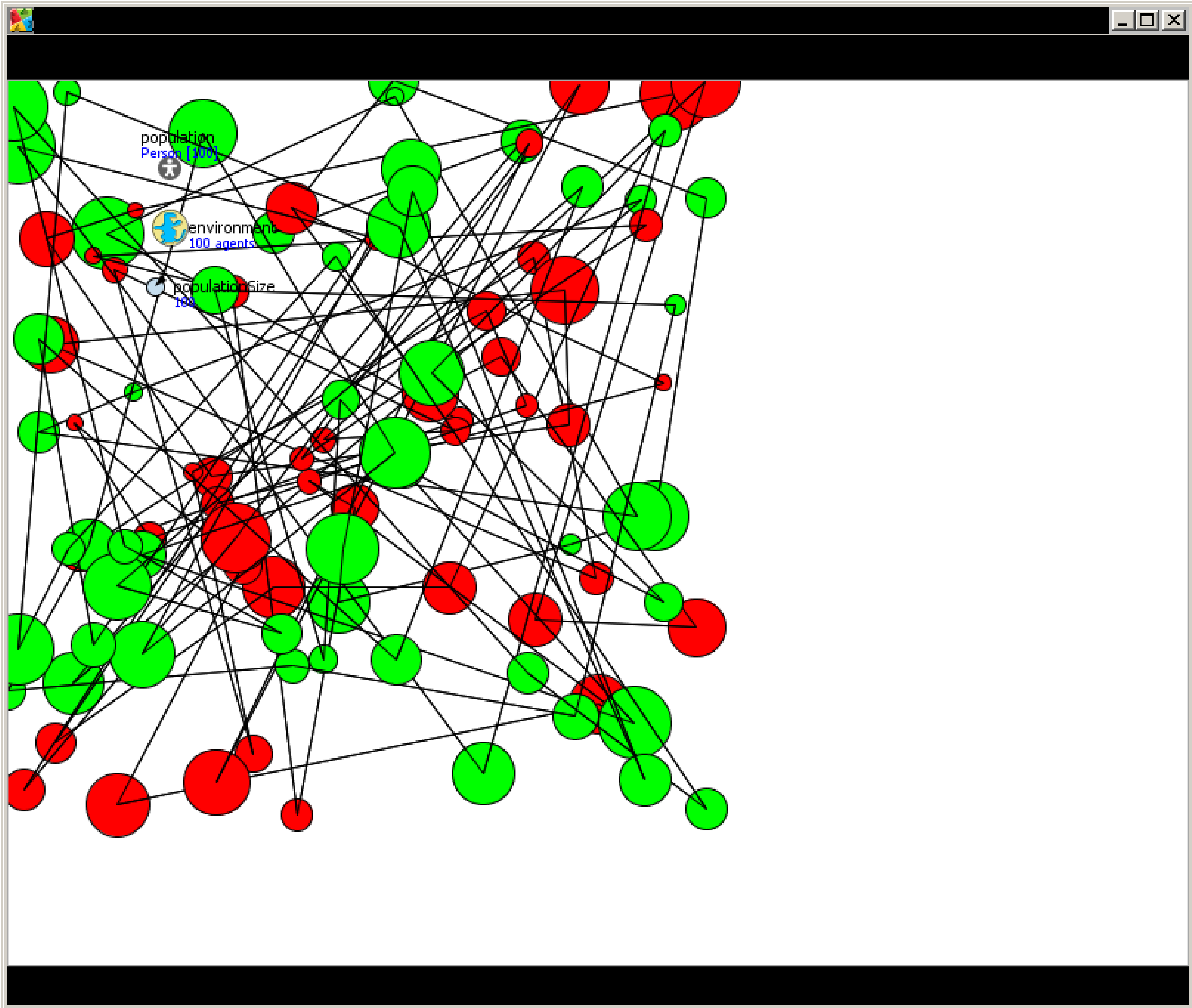
Property for Distance-Based Layout: Distance Threshold



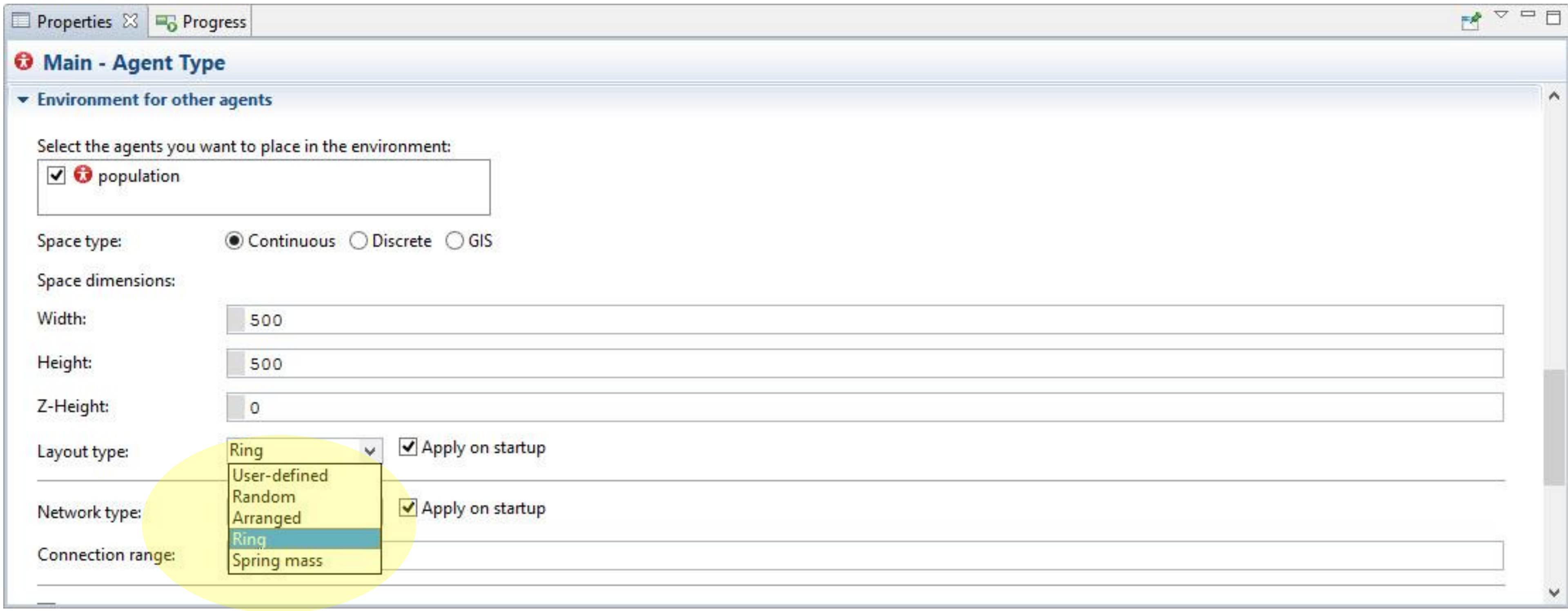
Purely Local Connections: Ring Lattice

- Purely **local** connectivity
  - Agents arranged in a ring
  - Connections by a given agent to some number of agents on either side of itself in the ring
- Slow propagation of infection (localized – no "short cuts" from one region to other regions)
- NB: Most naturally displayed with "Ring" "Layout type"

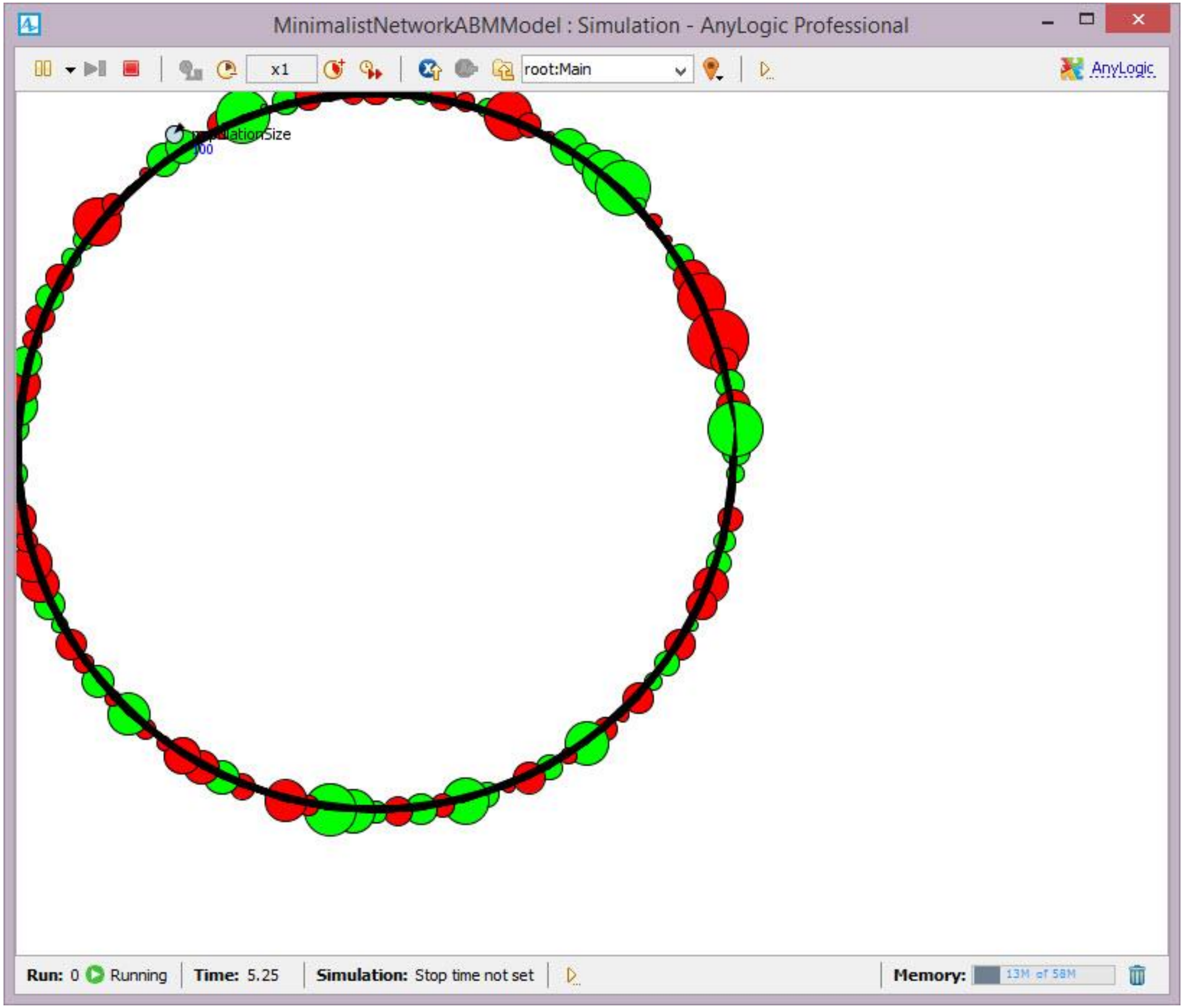
Ring Lattice – No Ring Layout



Ring Lattice – Choosing Ring Layout



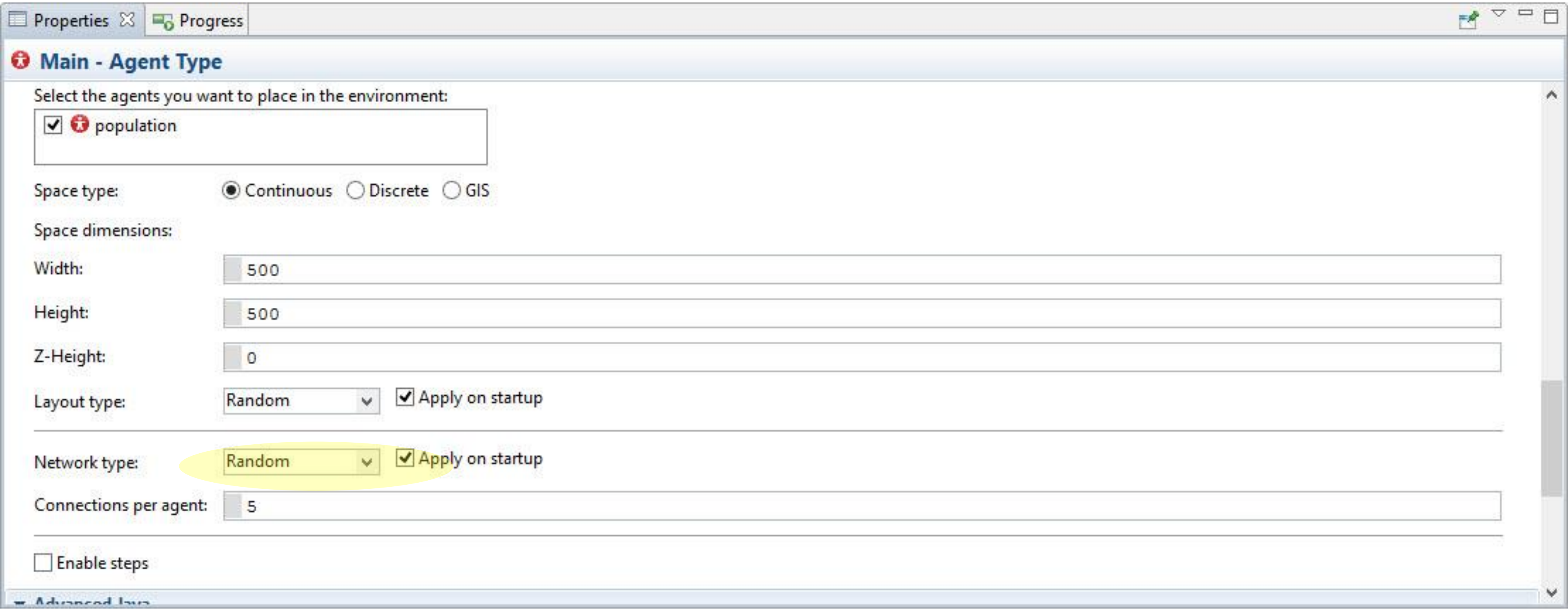
Ring Lattice Topology – With Ring Layout



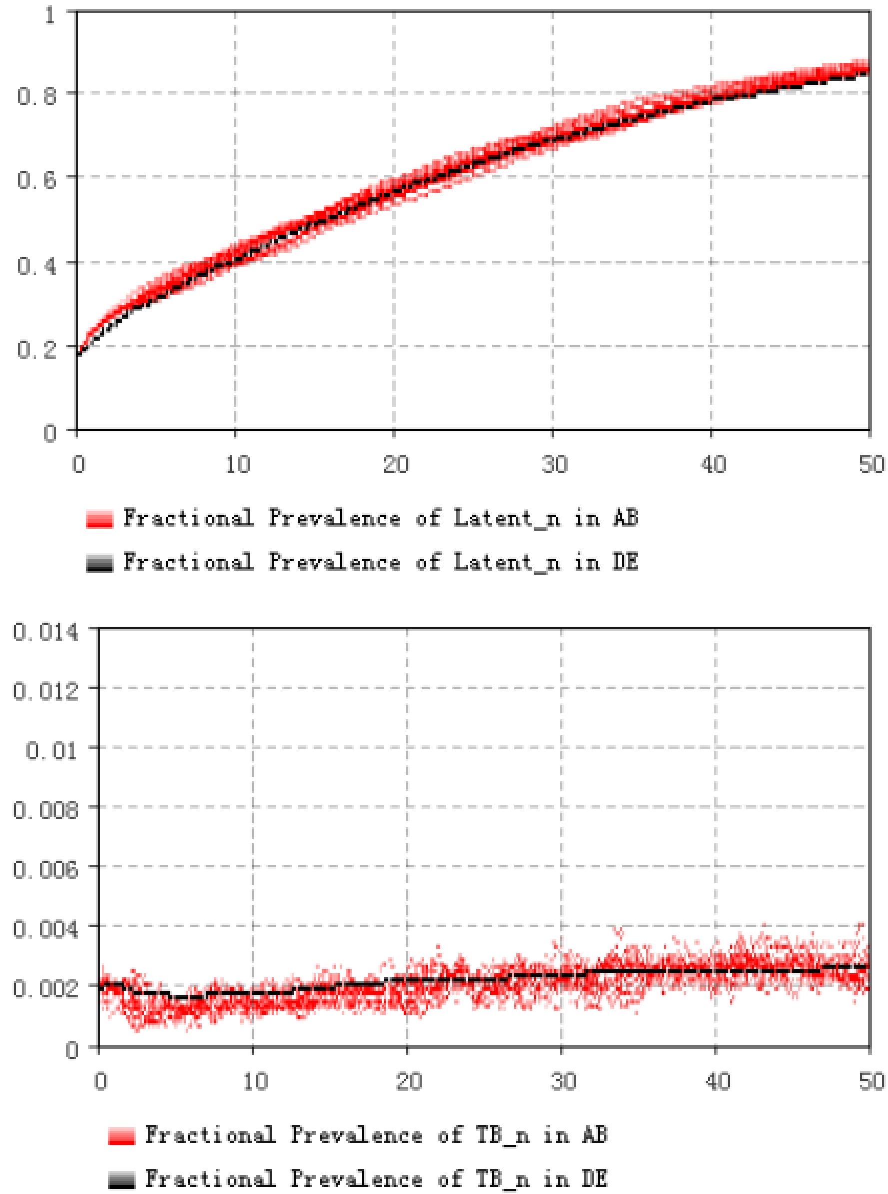
Global Connectivity: Poisson Random Networks

- In Poisson random networks (also called “random networks” or “Bernoulli networks”), any pair of nodes (A,B) exhibits the same chance of connection as any other pair of nodes
- This network type has no preference for any sort of “locality” (topological or spatial)
  - There is no more overlap in the connections of two neighbors than among two arbitrary nodes in the population

Global Random Mixing: Random Connections

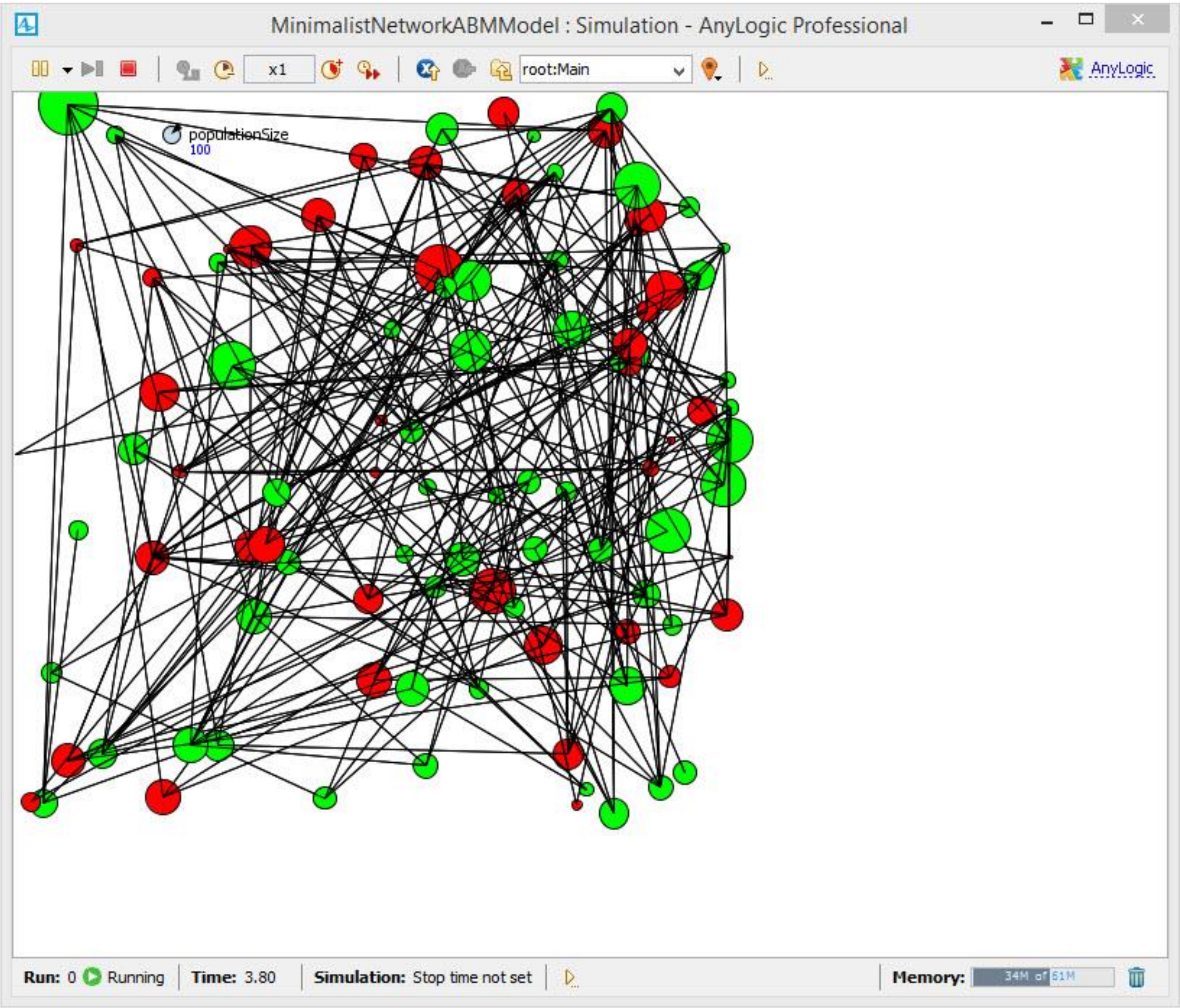


Connections over static Random networks can yield results very similar to what res from continuous, dynamic random mixing in an aggregate model



Random

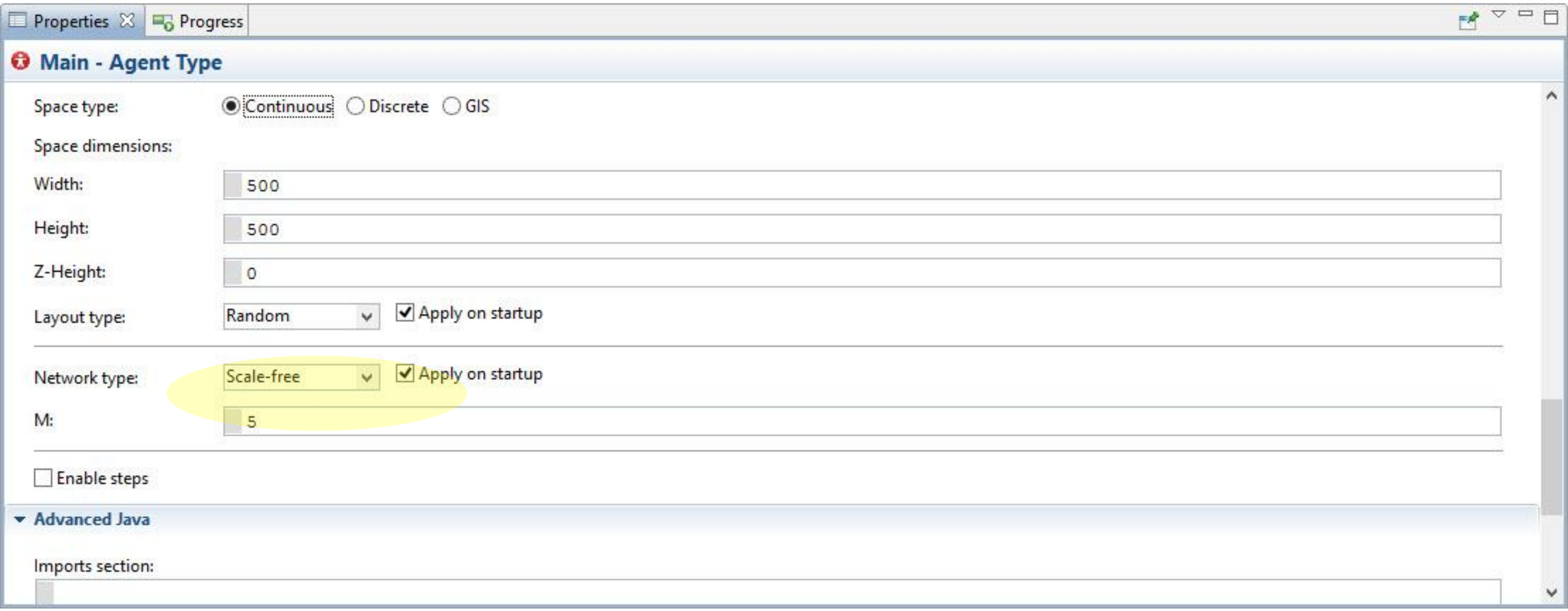
With Random Connections



Small World

- Small world networks represent a sort of "weighted combination" of
  - Ring lattice network (purely local connections)
  - Random network (mostly global connections)
  - The "Neighbor link fraction" in AnyLogic dictates what fractions of the connections are to the local neighbors (per ring lattice)
- Beware of the inconsistency in the definition of "Connections per agent" for small world networks
  - Off by a factor of two!

Scale-Free Network



Intuitive Plausibility of Importance of Heterogeneity

- Someone with high # of partners is both
  - More likely to be infected by a partners
  - More likely to pass on the infection to another person
- Via targeted interventions on high contact people, may be able to achieve great "bang for the buck"
- We may see very different infection rates in high contact-rate individuals

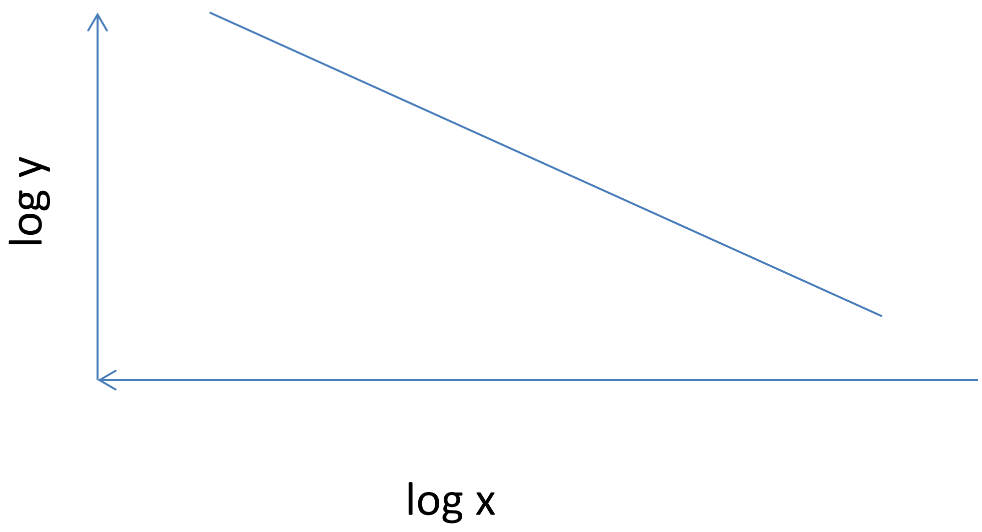
- **How to modify classic equations to account for heterogeneity? How affects infection spread?**

Scale-Free Networks

- A node's number of connections (a person's # of contacts) is denoted *k*
- Chance of having k partners is proportional to $k^{-\gamma}$.
- For human sexual networks, γ is between 2 and 3.5
  - E.g. if $\gamma=2$, likelihood having 2 partner is proportional to ¼, of having 3 is proportional to 1/9, etc.
- We will discuss scale free networks in a separate set of lecture slides

Recall: Power Law Scaling & Log-Log Graphs

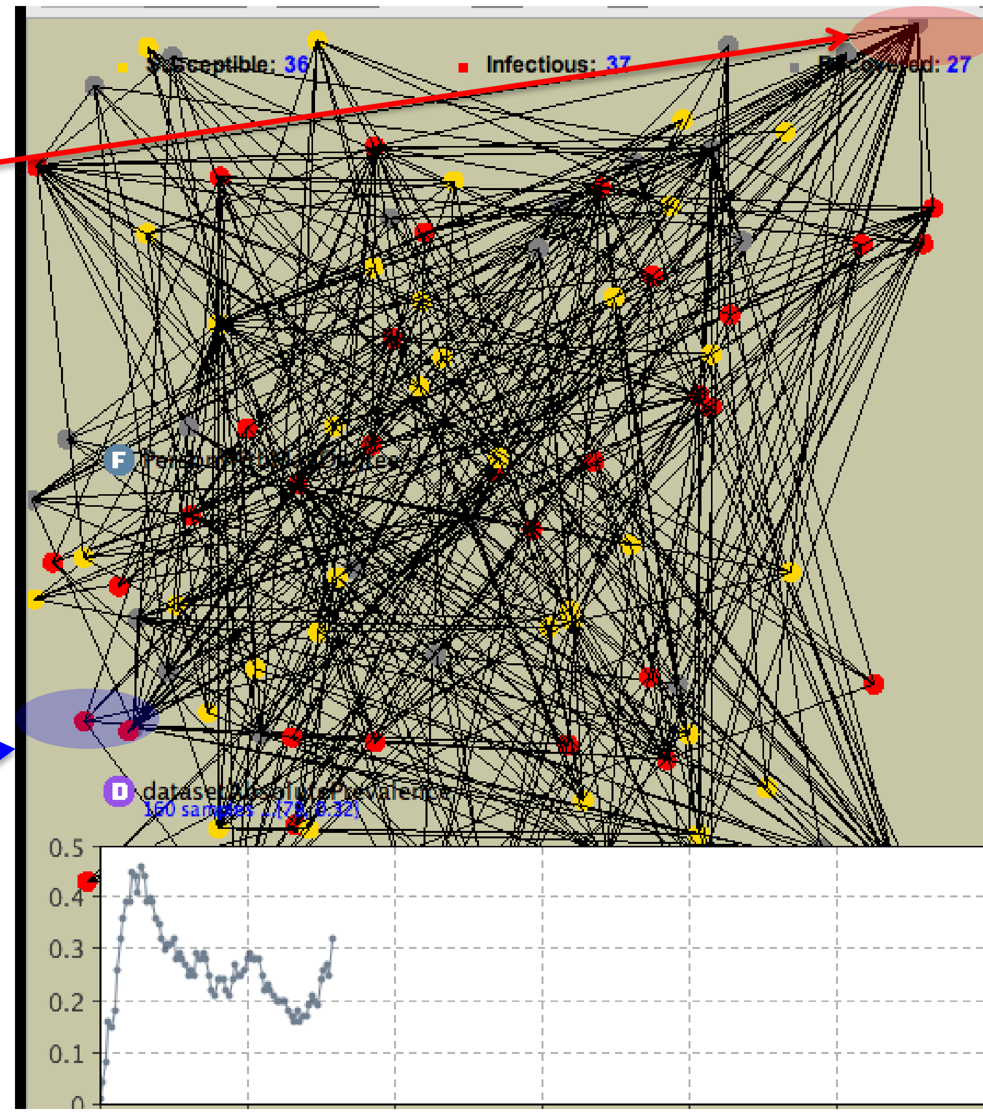
- $y=x^a$
- $\log y = a \log x$
- If x is negative, have something like



Scale-Free Network

High degree node

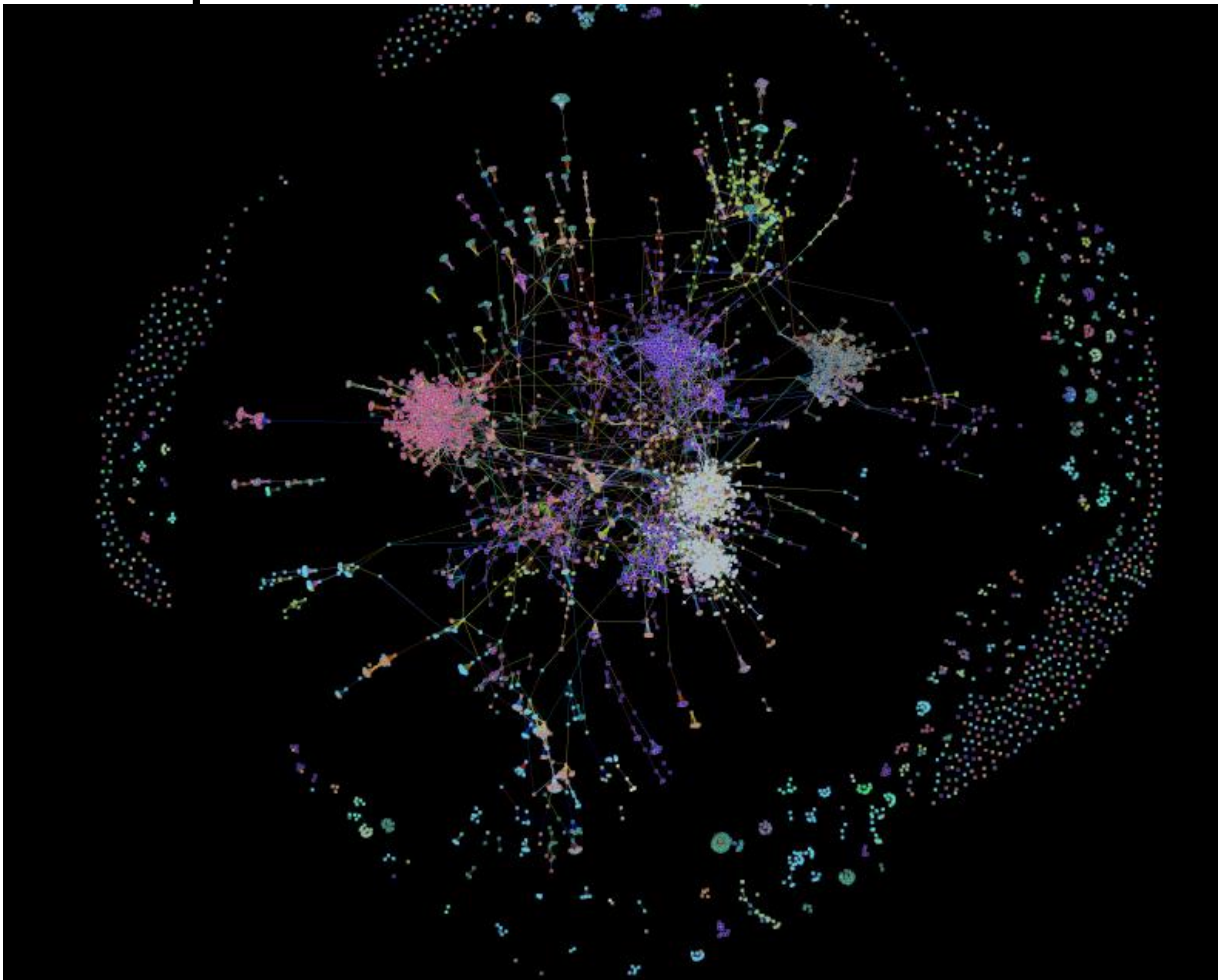
Low degree node



Interaction Between Network & Location 2

- In a Spring-mass layout, the nodes that are highly connected will tend to be clustered
- Here, we are determining the location based on the connectivity!

Example Network Substructure



General ABM Network Caveats

- In thinking about the effects of & tradeoffs between interventions, need to recognize that networks are emergent phenomena, driven by
  - Mobility patterns
  - Relationship formulation & dissolution
- Many networks are dynamic, but traditional measures rarely yield dynamic high temporal resolution data
- We typically have only partial information on network structure
- Often collected via a non-random sampling process
- Networks specific to definition of “contact”

Example: Contact Tracing Networks

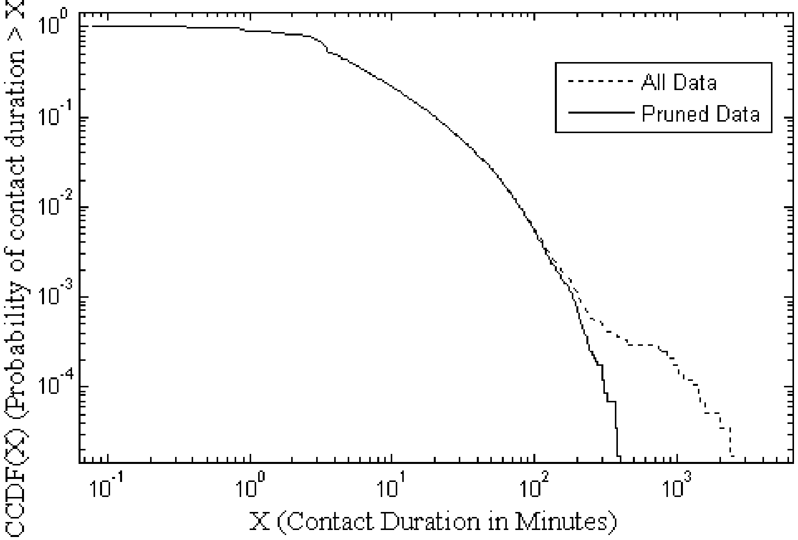
- These are produced by an asymmetric or biased contact tracing protocol
  - Uses definition of contact (e.g. needle-sharing incident, spending >8 hours in past 30 days, past or ongoing sexual relationship)
  - Perform tracing only under certain conditions
- Data at hand is likely collected over a substantial amount of time
  - The network may have changed during this time
- Unclear what this says about the network of the general population

AnyLogic Network Caveats

- Built-in networks are handy for routine tasks, but do not offer much flexibility e.g. preferential attachment, post-construction additions, etc.
- Constructing built-in networks can computationally expensive
- The "M" parameter in a Scale-Free network would not appear to be either classic parameters γ nor m (from Barabasi paper)
  - **Mean # of connections/person is approximately twice this value**
  - Number of connections per individual are often in discrete categories?
- NB: The "Small world" network uses a definition of connections/person inconsistent with those for other networks
  - Off by a factor of 2!

Network Dynamics in AnyLogic

- Observed networks are often dynamic over a wide range of timescales
- These dynamics can be very important to overall system dynamics.
- We can represent switching connections using
  – Removing a connection
  – Adding a new connection



Hashemian, M., Stanley, K., and Osgood, N. 2010. Flunet: Automated tracking of contacts during flu season. Proceedings of the 6th International workshop on Wireless Network Measurements (WiNMee 2010), 557-562, 6pp.

Automatically Wired Connections

- Predefined built-in (i.e. non-user-defined) AnyLogic network types can take care of “wiring in” a new node into an existing network
  - Just call *environment*.applyNetwork() to get the environment to “recalculate” the network – and thereby include the new node.

AnyLogic methods for Adding & Deleting Connections

- *agentA*.connectTo(*agentB*)
  - Connects *agentA* to *agentB*
  - NB: Connections are assumed to be undirected and symmetric (i.e. if *agentA* is considered to be connected to *agentB*, then *agentB* is considered to be connected to *agentA*)
- *agentA*.disconnectFrom(*agentB*)
  - Disconnects *agentA* and *agentB* from each other
- For more details and additional methods, see the slides for the *Networks* lecture

Useful Methods for Dealing with Networks

- *agentA*.getConnectionsNumber()
  - Gets count of connections associated with agentA
- *agentA*.isConnectedTo(*agentB*)
  - Return true if agentA and agentB are connected; false otherwise
- *agent*.getConnectedAgent(int *index*)
  - Returns the *index*th agent connected to agentA. Note: The first person is at index 0 (not index 1!)
- *agent*.getConnections()
  - Returns list (LinkedList<Agent>) of all connections of Agent *agent*. Can loop over this with e.g. a *for* loop