IONTW Reference Guide
Version 1.1

M. Drew LaMar* Winfried Just† Hannah Callender‡

Contents

*The College of William and Mary E-mail: mdlama@wm.edu

†Department of Mathematics, Ohio University, Athens, OH 45701 E-mail: mathjust@gmail.com

‡University of Portland E-mail: callende@up.edu

1 Infection Model Types

Infection Model	**gain-immunity**	**latent-period**	**end-infection-rate/prob**
SI	**Off**	**Off**	$= 0$
SIS	**Off**	**Off**	> 0
SIR	**On**	**Off**	> 0
SEI	**Off**	**On**	$= 0$
SEIS	**Off**	**On**	> 0
SEIR	**On**	**On**	> 0

2 Disease Transmission Parameters

- **infection-rate**: Rate of making effective contacts with an infectious host (β)
- **end-infection-rate**: Rate of loss of infectiousness $\left(\alpha = \frac{1}{\langle \tau^I \rangle}\right)$
- **end-latency-rate**: Rate of onset of infectiousness (γ)
- **infection-prob**: Probability of at least one effective contact with an infectious host over time Δt (Model parameter b). The formula relating the continuous infection rate β to the discrete probability b is given by

$$b = 1 - e^{-\beta \Delta t} \tag{1}$$

- **end-infection-prob**: Probability of loss of infectiousness over time Δt (Model parameter a). The formula relating the continuous end infection rate α to the discrete probability a is given by

$$a = 1 - e^{-\alpha \Delta t} \tag{2}$$

- **end-latency-prob**: Probability of onset of infectiousness over time Δt. This is derived from the continuous end latency rate γ using the formula

$$1 - e^{-\gamma \Delta t} \tag{3}$$

3 Numerical Simulation Parameters

- **model-time**: Denotes whether the model is a **Continuous** time model or a **Discrete** time model
- **time-step**: Time step used in discrete time models (Δt)
- **Discrete Approx**: Calculates discrete probabilites from continuous rates using equations (1)–(3)

4 Network Parameters

4.1 Network Creation

- **network-type**: Types of supported networks. See Section 5 for a descriptive list.
- **num-nodes**: Number of nodes (N)
- **lambda**: One of two possible parameters to vary properties of a specific network type. See Section 5 for its use in each type of network.
- **d**: One of two possible parameters to vary properties of a specific network type. See Section 5 for its use in each type of network.

4.2 Network Modification and Visualization

- **spawn-kill:** Specifies whether the user wants to *spawn* (create) or *kill* (remove) a node or link in the network. Used in conjunction with the **Node** and **Link** buttons.
- **Node:** When pressed, a node is created or destroyed (depending on selection of **spawn-kill**) when a user clicks in the **World** window. If **spawn-kill** is set to **kill**, the nearest node to where the user clicks is destroyed.
- **Link:** When pressed, the user can create or destroy links (depending on selection of **spawn-kill**) as follows: on the first click of the mouse in the **World** window, the closest node to this click is selected and set to white; on the second click of the mouse, a link is created or destroyed between the first selected node and the node that is closest in space to the second click.
- **Clear:** Clears the network, setting **num-nodes** to 0.
- **Randomize:** While pressed, performs random edge swaps between nodes without changing the degree sequence. Useful for exploring results of different network realizations of fixed degree sequences.
- **Metrics:** Displays network and simulation metrics in the **Command Center**. Current metrics include:
  - Simulation parameters
    - `<tau>`: Mean time of infectiousness $\langle \tau^I \rangle$
    - `R0`: Basic reproductive ratio R_0
    - `Maximum number of simultaneous infections`
  - Network parameters

* `Mean degree`: If the degree of node i is denoted by k_i, then this is given by

$$\frac{1}{N}\sum_{i=1}^{N} k_i.$$

* `Edge density`: This is the number of edges $|E|$ divided by the number of all possible edges, given by

$$\frac{2|E|}{N(N-1)}.$$

* `Clustering coefficient`: If $N > 1$, computes the mean node clustering coefficient over all nodes. For a node i, let $\mathcal{N}_1(i)$ denote the set of nodes that link to i. Let $tr(i)$ denote the number of links $\{j_1, j_2\}$ such that $j_1, j_2 \in \mathcal{N}_1(i)$. The node clustering coefficient is given by

$$C(i) = \begin{cases} \frac{2tr(i)}{k_i(k_i-1)} & \text{if } k_i > 1, \\ \frac{2|E|}{N(N-1)} & \text{if } k_i \leq 1. \end{cases}$$

  The network clustering coefficient is then given by

$$C = \frac{1}{N}\sum_{i=1}^{N} C(i).$$

* `Normalized clustering coefficient`: If $N > 1$, computes the mean normalized node clustering coefficient. The normalized node clustering coefficient is given by

$$C_{norm}(i) = \begin{cases} C(i)\frac{N(N-1)}{2|E|} & \text{if } L > 0, \\ 1 & \text{if } L = 0, \end{cases}$$

  and thus the normalized network clustering coefficient is given by

$$C_{norm} = \frac{1}{N}\sum_{i=1}^{N} C_{norm}(i).$$

* `Number of connected components`
* `Largest component (as proportion of network)`
* `Average path length in largest component`: If $\{i_1, \ldots, i_n\}$ denotes the nodes in the largest component, then the average path length in the largest component is given by

$$\frac{2}{n(n-1)}\sum_{j>m} d(i_j, i_m),$$

  where $d(i_j, i_m)$ is the number of edges in the shortest path between i_j and i_m.

    * `Diameter of largest component`: The diameter of the largest component is the maximum shortest path in the largest component, i.e.

$$\max_{j>m} d(i_j, i_m),$$

where, as above, $d(i_j, i_m)$ is the number of edges in the shortest path between two nodes i_j and i_m in the largest component.

- **Labels**: Toggles node labels on and off.
- **Scale**: Scales up the network to fill the world window. Useful in conjunction with **Spring**, which tends to contract everything down.
- **Spring**: Useful to "clean up" network. Simulates links as springs. While this is pressed, the network will approach an equilibrium configuration. We recommend using **Spring** with the fastest setting of the speed control slider. Particularly useful for trees (although fun to use with all network types).
- **plot-metric**: In conjunction with the **Network Metrics** plot window, selects the type of information to display. Current choices include:
  - **Degree Distribution**: Distribution of node degrees
  - **Clustering Coeffs**: Histogram of node clustering coefficients
  - **Normalized Coeffs**: Histogram of normalized node clustering coefficients
  - **Shortest Paths**: Histogram of shortest paths in largest component
  - **Probability Distribution**: Distribution corresponding to the **Custom Distribution** network type.
- **Update**: Updates the **Network Metrics** plot window to plot the metric chosen in **plot-metric**. Note that when loading a distribution from file, **plot-metric** is automatically set to **Probability Distribution** and the distribution is displayed. Also, any change in the currently displayed network that changes the number of nodes or links updates automatically the **Degree Distribution** plot.

5 Network Types

All networks are undirected and *simple*, i.e. no self-loops or multi-edges.

- **Complete graph**

  Parameters: **num-nodes** (N)

  Description: Graph K_N consisting of N nodes with edges between every pair of distinct nodes.

- **Empty graph**

  Parameters: **num-nodes** (N)

  Description: Graph $\overline{K}_N$ consisting of N isolated nodes with no edges.

- **Erdos-Renyi**

  Parameters: **num-nodes** (N), **lambda** (λ)

  Description: Implements the version of Erdős-Rényi random graphs where

  $$p = \frac{\lambda}{N-1}$$

  is the probability that each possible edge is included in the graph, and λ is the expected mean degree.

- **Nearest neighbor 1**

  Parameters: **num-nodes** (N), **d**

  Description: Creates a graph with N nodes in a ring, connecting the **d** nearest neighbors to each side of every node. This creates a particular instance of a 2**d**-regular graph.

- **Nearest neighbor 2**

  Parameters: **num-nodes** (N), **d**

  Description: Creates an $n \times m$ grid of nodes, where $N = nm$ with m the largest integer factor of N. The integer **d** connects the nodes in a nearest-neighbor fashion as follows:

  - **d** $= 0$: Empty graph
  - **d** $= 1$: Each node connects to the nodes that are to the north, south, east and west.
  - **d** ≥ 2: If we denote the index of a node by (i, j), then this node will connect to all nodes with indices $(i \pm k, j \pm l)$, where $0 \leq k, l \leq$ **d**-1.

- **Small world 1**

  Parameters: **num-nodes**, **lambda**, **d**

  Description: Creates the union of an **Erdos-Renyi** graph with parameters **num-nodes** and **lambda** and a **Nearest-neighbor 1** graph with parameters **num-nodes** and **d**.

- **Small world 2**

  Parameters: **num-nodes**, **lambda**, **d**

  Description: Creates the union of an **Erdos-Renyi** graph with parameters **num-nodes** and **lambda** and a **Nearest-neighbor 2** graph with parameters **num-nodes** and **d**.

- **Preferential attachment**

  Parameters: **num-nodes** (N), **lambda** (m_0), **d** (m)

  Description: Create a graph using a modified form of the preferential attachment algorithm. We start with an initial graph $G^{(0)}$ as the complete graph on m_0 nodes. We then perform the following iteration over $i = 0, \ldots, N - m_0$:

  1. Let $\bar{k}^{(i)}$ be the vector of degrees of all nodes in the graph $G^{(i)}$
  2. Add a new node and connect this node to m nodes in $G^{(i)}$, choosing these m nodes randomly in proportion to their degrees in $\bar{k}^{(i)}$.
  3. Increment i by 1, and repeat 1-2 until number of nodes is N.

  Note: This algorithm is animated in the **World** window for illustrative purposes. If you would like to speed up the process, you can either unselect **view updates** or adjust with the speed slider.

- **Generic scale-free**

  Parameters: **num-nodes** (N), **lambda** (λ)

  Description: Creates a random scale-free graph with degrees drawn from the distribution

  $$f(k) = \frac{c}{k^{\lambda}},$$

  where $k = 1, \ldots, N - 1$ and $\lambda \geq 1$. The value c is chosen so that $f(k)$ sums to 1.

- **Spatially clustered**

  Parameters: **num-nodes** (N), **lambda** (λ)

  Description: Creates a spatially clustered graph by first assigning N nodes x and y coordinates uniformly at random in the **World** window. While the edge density of the graph is less than λ, nodes are chosen at random and connected to their nearest neighbor.

- **Random regular**

  Parameters: **num-nodes** (N), **lambda** (λ)

  Description: Creates a random graph with N nodes such that every node has degree λ.

- **Regular tree**

  Parameters: **lambda**, **d**

  Description: Creates a rooted tree with depth **lambda** and **d** branchings per interior node.

- **Custom distribution**

  Parameters: None

  Description: When a degree distribution has been pre-loaded (see **Load** below) and this network type is selected, each press of the **New** button will draw a new degree sequence from this distribution. You can randomize the graph while keeping the degree sequence fixed by pressing the **Randomize** button. You can move between the specific network's degree distribution and the custom distribution by changing **plot-metric** and pressing **Update**.

6 Setup & Go

- **New**: Creates a new network based on the **network-type** menu and corresponding network parameters **lambda** and **d**, if appropriate. *When using BehaviorSpace, the command **new-network** performs the same actions.*

- **Last**: Resets the node states to the previous initial conditions. *When using BehaviorSpace, the command **last-init** performs the same actions.*

- **Go**: Run the simulation. Press again to stop a simulation. *When using BehaviorSpace, the command **go** performs the same actions.*

- **Defaults**: Sets all simulation, disease and network parameters to defaults.

- **Load**: Loads information from file. The types of information that can be loaded include:

  - Degree sequence (see `degree.txt` for an example)
  - Degree distribution (see `distribution.txt` for an example)
  - Network (this is "Network Parameters" as referred to in this document, as well as nodes, links and node states – see `sample-network-detailed.txt` for an example with a detailed format, and `sample-network.txt` for an example with a simple format).
  - Parameters (this is "Simulation and Disease Parameters" as referred to in this document – see `parameters.txt` for an example).
  - All (Networks and Parameters – see `all.txt` for an example).

  Note: File text is sensitive to both case and whitespace.

- **Save**: Saves information to file. The types that can be saved include:

  - Network (this is "Network Parameters" as referred to in this document, as well as nodes, links and node states – see `sample-network-detailed.txt` for an example).

  - Parameters (this is "Simulation and Disease Parameters" as referred to in this document – see `parameters.txt` for an example).
  - All (Networks and Parameters) – see `all.txt` for an example).

6.1 Initialize States

- **set-state-to**: Can be **Infectious** or **Removed**. Specifies the state you would like to initialize nodes to upon pressing the **Set** button.
- **set-state-by**: Specifies the way in which you would like to change the node states. Choice include:
  - **Number of nodes**: Needs to be an `integer`.
  - **Fraction of nodes**: Between 0 and 1.
  - **Vector from input:** Specify node labels in list format (e.g. `[0 1 10 16]`).
  - **Vector from file:** File should have one line with a list of node labels (e.g. `[0 1 10 16]`).
- **num/frac:** Stands for "number or fraction," depending on the choice of **set-state**. This is how you specify the number or fraction of nodes to infect or immunize.
- **min-deg:** Specifies the minimum degree for the population of nodes to apply the **set-state** algorithm to.
- **Set:** Applies the state initialization specified *to all susceptible nodes* with all of the above choices. By performing state initializations in serial, you can, for example, immunize 50% of the susceptible nodes and infect 10% of the remaining susceptible nodes.
- **Reset:** Sets the state of all nodes to susceptible.
- **Select:** When pressed, a user can cycle through the states for a node by clicking on the node.
- **auto-set:** When set to **On**, every network created with the **New** button will automatically press the **Set** button.

7 Helpful Commands

Description	IONTW Commands
Report number of susceptibles	`count turtles with [susceptible?]`
Report number of latents	`count turtles with [latent?]`
Report number of infectious	`count turtles with [infectious?]`
Report number of removed	`count turtles with [removed?]`
Report mean degree	`mean [count link-neighbors] of turtles`
Report time	`ticks`
Change label for nodes with degree 3	`ask turtles with [count link-neighbors = 3][set label "label"]`
Create new network	`new-network`
Make one individual susceptible	`ask n-of 1 turtles [become-susceptible]`
Make one individual latent	`ask n-of 1 turtles [become-latent]`
Make one individual infectious	`ask n-of 1 turtles [become-infectious]`
Make one individual removed	`ask n-of 1 turtles [become-removed]`
Make one susceptible removed	`ask n-of 1 turtles with [susceptible?][become-removed]`
Vaccinate individuals from file	`ask turtles-from-file "vaccinate.txt" [become-removed]`
Infect individuals from file	`ask turtles-from-file "infect.txt" [become-infectious]`
Remove individual with degree > 6	`ask n-of 1 turtles with [count link-neighbors > 6][become-removed]`
Remove everyone with degree > 6	`ask turtles with [count link-neighbors > 6][become-removed]`