Simulator Project – Falsification Detection

# Description and Goals

This project aims to simulate pharmaceutical supply chains in a low- or middle-income country (LMIC) that are subject to insertions of substandard and/or falsified products (hereby referred to as “falsified” products). The supply chain of a single pharmaceutical product (e.g., amoxicillin) is characterized through a network of nodes and arcs, where falsifier product originates from a single falsifier source node. Entities such as importers or outlets may possibly obtain product from these falsifier nodes under different circumstances or configurations. Drug regulatory agencies (DRAs) have the ability to sample and test products from different entities according to testing policies and procurement budgets.

The general aim of this project is summarized as follows: “Under what types of network configurations and falsification contexts can different testing policies most ably detect the underlying structures of falsification?” Falsification structures need to be ascertained using the following pieces of information:

* Sample collection results
  + Importer name
  + Batch in-country arrival date
  + Pass/Fail diagnostic reading
* Reports of stockouts at different entities – either when arriving to collect a sample, or by other means

Summarily, this project will explore some of the following questions:

* Can the location of “bad actors” be detected in the network, and with what degree of confidence? Under what circumstances?
* How do the testing data appear under temporal network changes, i.e., the occurrence of unfit batches of a product that is otherwise fit for consumption?
* How can sampling data and supply chain structure be integrated into decision-making policies?
* What happens as the network expands/contracts in complexity?
* Which testing policies are most/least effective (at all)?

# Simulation Model Schematic

(See “Falsification\_Simulator.py” for the Python code running this model.) The simulation model utilizes a graph-based structure to represent the pharmaceutical supply chain. The primary objects are “entity nodes,” signifying manufacturers, importers, and pharmaceutical outlets, and “drug packets,” signifying a single dosage of a particular pharmaceutical product. Entity nodes are generated using the following lists:

* **Root Nodes:** Signifies the entering of a drug into the supply chain. Assume the root node always has enough supply to meet the demand of all nodes it is supplying (e.g., global supplier).
* **End Nodes:** The end of the movement of a drug in the supply chain.
* **Intermediate Nodes:** Currently in the middle of the supply chain. The drug is not at a root node nor an end node

The simulation model takes into account the demands of each non-root node from its reorder points and reorder amounts. While also accounting for each end node’s preferred suppliers, an effective demand schedule can be developed. The simulation tracks the entire transition life of each drug packet, so it can be known if drugs originated from a falsifier node.

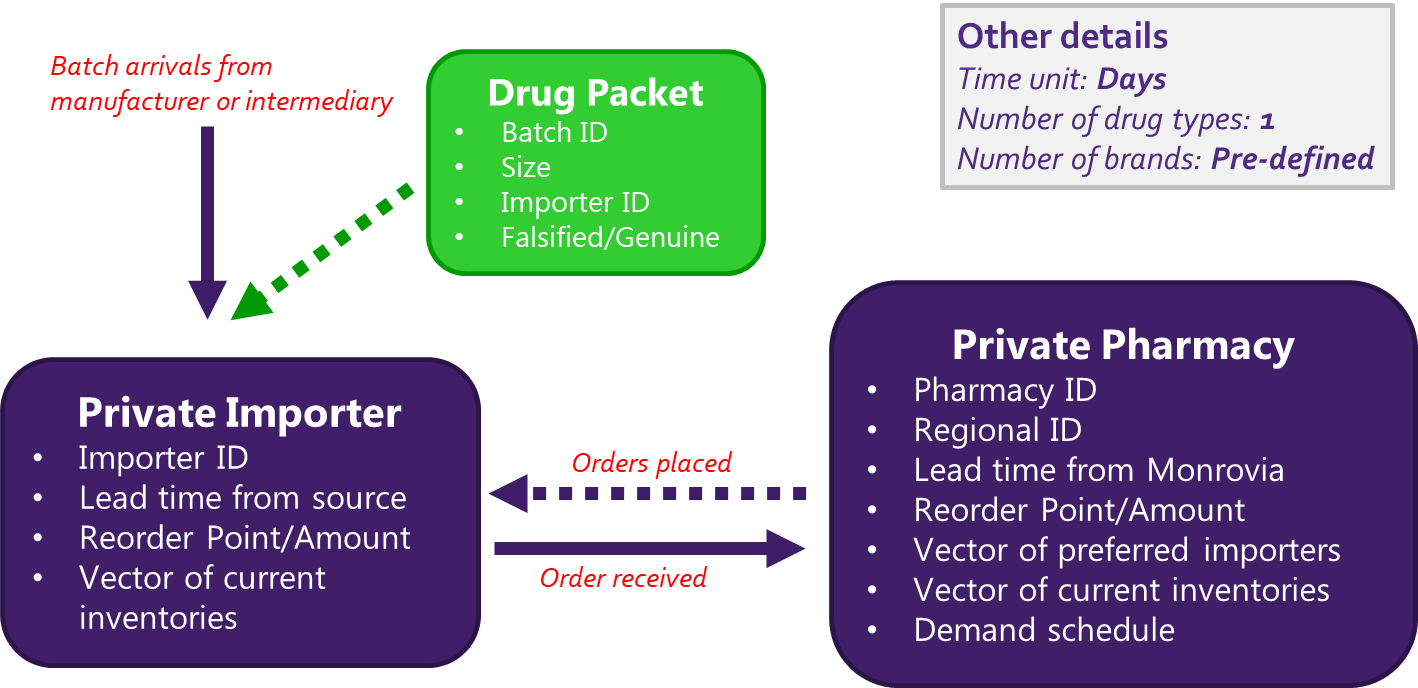
By taking into account the list of end nodes and sampling budget, an estimate of the percentage of how many falsified drugs at each node within a supply chain can be developed.

The simulation timeline is as follows:

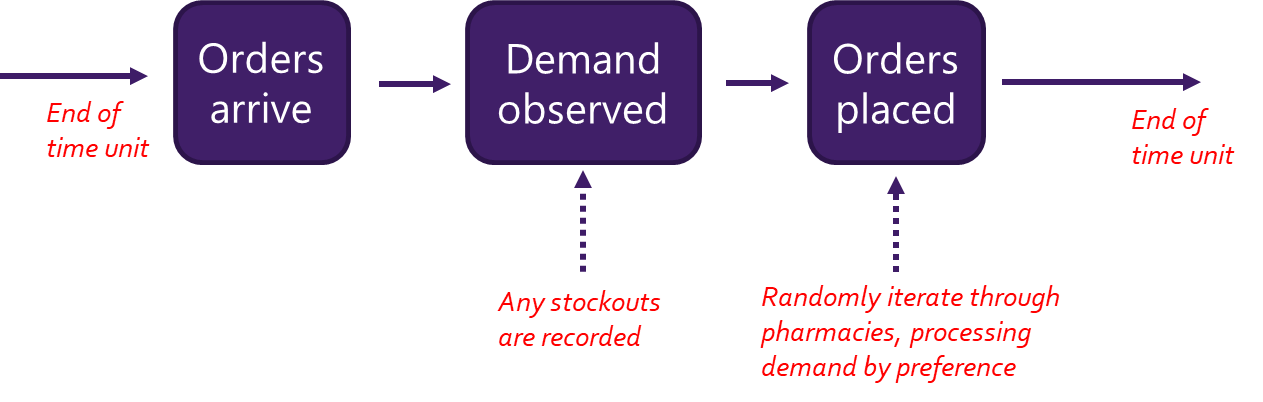
* Input files are read into the system. These input files provide:
  + The list of nodes
  + Arc preferences
  + Arc lead times
* Parameters are set such as the number of simulation days, sample budget, and the drug testing policy to use
* Data from the input files are used to create lists of root, intermediate, and end nodes
* Based on the testing policy selected and arc preferences, testing schedules are developed
  + Possible testing policies include:
    - a deterministic end node policy
    - a deterministic end node policy with multiple tests per day set via the sampling budget
    - a randomized end node policy with multiple tests per day set via the sampling budget
* Each day of the simulation, the intermediate and end nodes process arriving orders by updating their drug packet consumption records, demand records (satisfied vs stocked out units), and time before the next inventory arrival
* When intermediate or end nodes have inventory levels at or below their reorder points, they place an order
* After the simulation runs through all of the days, data is outputted regarding the following:
  + Root node summary statistics→ the consumption of each root node is provided followed by its consumption percentage compared to the other root nodes
  + Intermediate node summary statistics→ For each intermediate node, the amount of satisfied demand followed by stockout demand is reported. The stockout percentage and inventory available for each node are also shown
  + End node summary statistics→ For each end node, the amount of satisfied demand followed by stockout demand is reported. The stockout percentage and inventory available for each node are also shown
  + Testing summary statistics
  + Overall statistics→ The overall simulation run time, total tests, total falsified drugs found, and total stocked out found is shown

The following graphics depict the different structural logic of each element of the simulation.

## Overview of the flow of products (“Drug Packets”)



## Timeline at the pharmacy outlet (end node) level



## Model Parameters

The following is a list of parameters that can be modified within the model:

* Reorder points/amount for different entities (i.e., internal policies)
* Lead times and global stockout rates (i.e. external factors)
* Underlying network structures
  + Number of importers that can be accessed by each outlet
  + Number of entities engaging with the falsifier node
  + Overall size/complexity
* Trigger sensitivities for entities opting to source from the falsifier node
* Testing policies and budgets

# Liberian Base Model

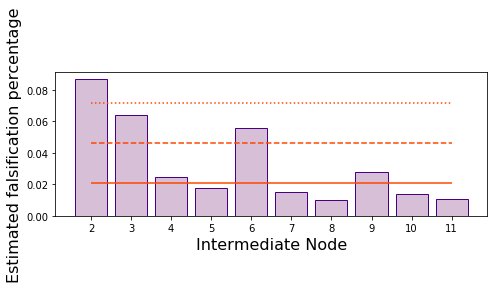
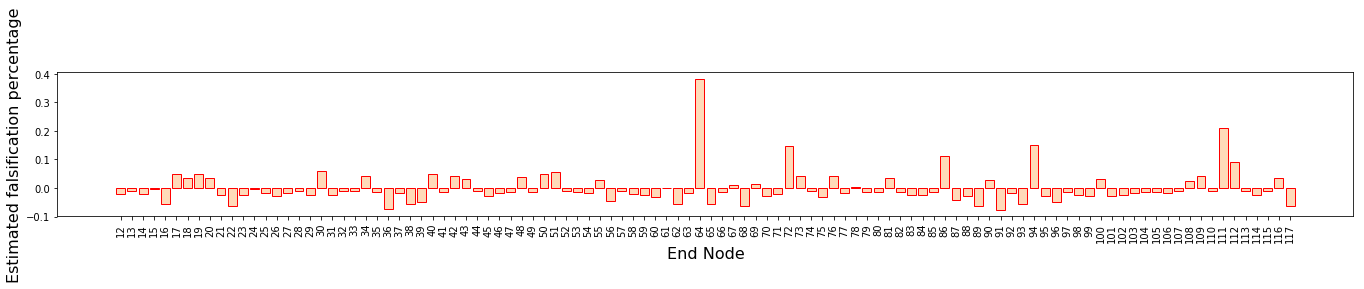
We make some assumptions in generating the Liberian base model, which acts as our best estimate of the scale and flow of Liberia’s private pharmaceutical supply chain for its 15 counties. These various assumptions may be adjusted for different configurations and scenarios:

* **1 falsifier node; 1 global manufacturer/supplier**: These are the two source nodes for all drug packet objects. Intermediate nodes (importers) may source from either node, while end nodes (pharmaceutical outlets) source either from intermediate nodes or the falsifier node.
* **10 importers**: During the September 2019 visit, it was established that there are roughly 10 major private importers of pharmaceuticals in Liberia that would provide the large majority of drugs for the private supply chain (especially antibiotics). This value could be adjusted for different pharmaceutical classes (doxycycline, ampicillin, etc.).
* **106 end nodes**: Currently we are only sure that there are 10 registered pharmacies in Bong County. Using population estimates, we extrapolate this knowledge to other counties, presuming that a similar pharmacy-per-capita ratio exists, which brings us to 106 total pharmacies in Liberia.
  + Ideally, we will have the exact number of registered outlets at some point.
* **Average demand per region**: The 2018 WHO Report on Surveillance of Antibiotic Consumption shows each of the surveilled African nations (4 total, excluding Liberia) consuming around 5 daily dosage XXx per 1000 inhabitants of beta-lactam antibacterials/penicillins. Amoxicillin falls under this umbrella and constitutes the most commonly consumed antibiotic in Liberia (amoxicillin signifies about 25% of this category in Burundi). Without differentiating, and assuming that we’re looking at all beta-lactam antibiotics in our simulation, we use an assumption of 5 days per treatment/blister pack to approximate a 1 blister pack/1000 inhabitants per day in Liberia.
  + These approximations can be greatly improved upon receiving importer information with respect to specific drugs, like amoxicillin, doxycycline, etc.
* **Average demand per pharmacy**: Once the demand per region is established, we can use the estimated pharmacy numbers to establish how much average demand should be experienced by each pharmacy. We assume pharmacies in a particular region see the same average demand.
* **Demand variability**: Using a target coefficient of variation of 0.5 and an assumed uniform distribution (lacking any other information on typical demands - Poisson would be just as reasonable, too), we can backwards solve the necessary upper/lower bounds on the uniform distribution to garner this desired CoV.
* **Lead time from Monrovia per region**: These values are best guesses from personal knowledge of round trips to and from each county of Liberia to the capital, Monrovia.
* **Inventory policies**: For the base model end nodes we set the reorder point equal to a 90% service rate and the reorder level equal to 15 days of mean demand. For intermediate nodes we use a 90% service rate relative to first-preference customers, and 60 days of mean demand.

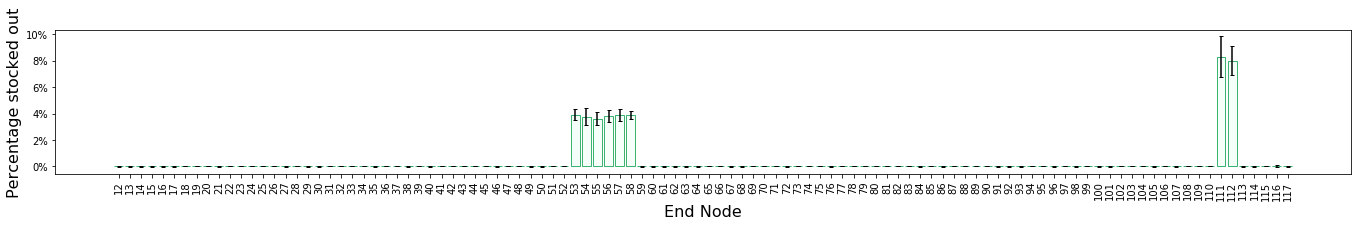
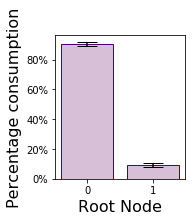
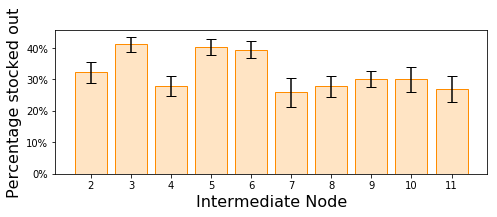
# Simulation output examples

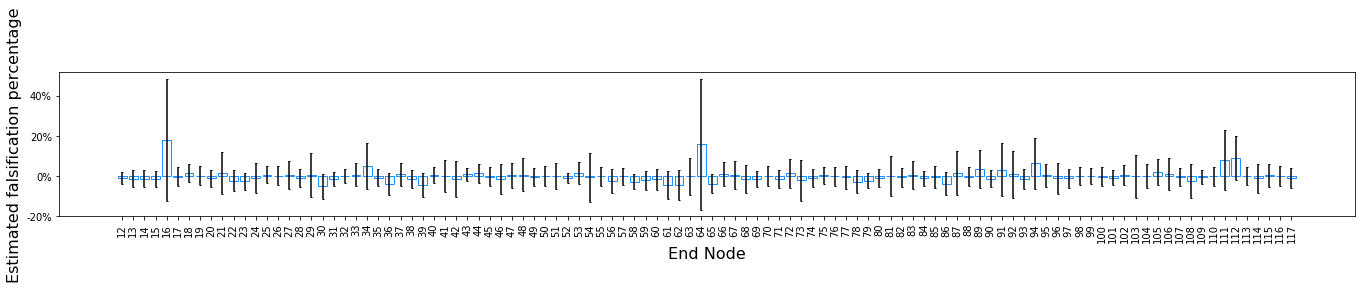
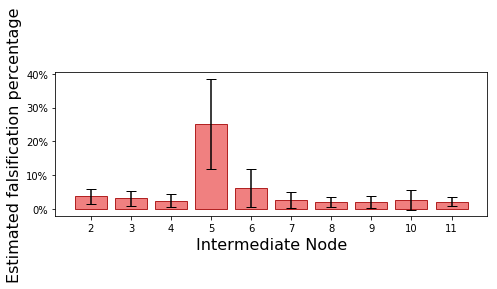
The following is a collection of different configurations and results that have been produced so far.

*Base Model –* 400 simulation days, 5 deterministic samples collected per day

*Problem Intermediate Node –* 1 intermediate node distributes SF batches with probability 0.2





# Building this model elsewhere

The principal drivers of this simulation model are the sampling data collected at different points in the system and an idea of the overall supply chain structure. Heightened information around each of these drivers will result in a more useful model. Although assumptions for any piece of data can help bridge gaps, the following is a list of different pieces of information that would help the simulation best emulate reality:

* **Supply chain structure**
  + Importer data
    - Inventory policies
    - Lead time to order reception
    - Size + date of shipments received
    - Demand data
    - Stockout prevalence
    - Procurement criteria for different manufacturers
  + Outlet data
    - Inventory policies
    - Lead time to order reception
    - Size + date of shipments received
    - Demand data
    - Stockout prevalence
    - Procurement criteria for different importers
    - Regional disease prevalence
* Detection capacity
  + Potential sampling plans
  + Regulatory actions
  + Sensitivity/specificity of screening diagnostic

# Moving Forward

The project will progress under the following steps:

1. Identify some base models and look at different output patterns
   1. What do different configurations look like with respect to the data we would observe?
   2. What is the effect of LT shocks? (Like when the southeastern region gets cut off from Monrovia)
2. How can sampling data be integrated with knowledge of the supply chain structure to produce estimates of where falsified product is emanating from (importers vs. outlets)?

# To-Do’s

*A list of items to complete:*

1. Put a smaller falsifier “order” amount in the “root” section of the ‘MakeOrder’ method
   * Should be triggered by the ‘current supplier’ node having the ‘FALSIFIER’ label
   * Potentially triggered by the current number of days stocked out, as well
   * Order amount is r/2
   * Record the number of triggers activated per entity
2. Generate ability to run batch files that vary different parameters
3. Put the “batch consumption rate” statistics in the simulation output
4. Streamline testing/sampling process into its own module
5. Implement "sandy" checks to ensure things are running smoothly without errors, outliers, etc.
6. Generate a “scratch folder”
7. Put the printed output logic into a module
8. Fix output reader for testing results; needs to include transition matrix estimates
9. Develop a procedure for what would be needed to build a simulation model in a much larger country like Nigeria
10. Put together a description of how the simulation model works
11. p\_hat estimates, matrix calculations, etc., need to be made robust to the possibility that not all nodes are tested.
12. Modifier/wrapper function that can change network structures, etc.
13. ~~Create output reader module for output dictionaries~~
14. ~~Put the warm-up logic into its own module~~
15. ~~Add sensitivity/specificity to testing results~~
16. ~~Add warm-up period~~
17. ~~Have multiple (100?) long warm-up periods that we sample from for each replication (as opposed to generating warm-up periods each time)~~
18. ~~Run multiple replications as opposed to long-run simulations~~