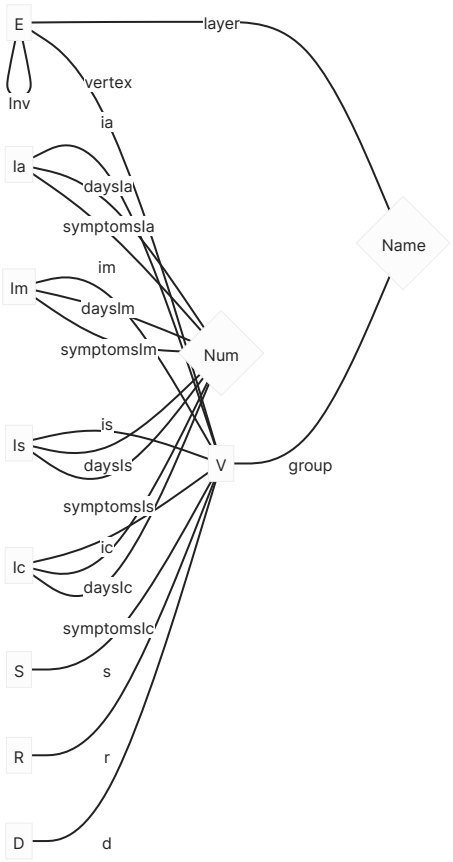
Current Code

Structure

- Update rules and schema are located in Complexvid.jl, test schedules and runs are in test.jl

Schema



Rules

Initial Infection Rule

- rule = initial_infect_r
  - Uses if_close_avg and query_vertices
- Automatically infects as mild
- **Need to implement random sampling from lognormal distribution with mean=1.621 and sigma=0.418. Right now it's just a random int**

Increment infected days

- rule = increment_r
- adds 1 to dayslm and substracts 1 from symptomslm

Check if it's time to show symptoms

- rule = checkI0
  - Uses is_symptom_day and q_im

Appearance of symptoms (mild infected)

- rule = symptoms_r
  - Removes edges from Im showing symptoms *except* for "home" edges

Check if infected for 20 days

- rule = i20

Flip for recovery

- rule = recoveryFlip
- Depends on when the Im started showing symptoms

Recovery

- rule = recoverMildApp

Schedules

- initial_infection_sched
  - A schedule for infecting an initial person given a graph populated with nodes and edges
- sched_rec
  - A schedule for looping through mildly infected people, incrementing infected days, starting symptoms, flipping for recovery, and recovering appropriately

Need to do

- Fix the random intialization of symptom days
- Create rules and schedule for initial edge connections and weights
- Create rules and schedule for contact infection between S and I
- Possibly generalize to non-mild Infected states as well
- Possibly implement "action" days (see below)

Model steps (may be incorrect)

Initialization

- Create nodes, assign ages according to age distributions

- All nodes start in "S" state with 0 days infected
- Create edges and assign weights (need to clarify this procedure, code is here)
- Calculate the average per-node degree of the graph.
  - While current infected $<$ initial infected (default 1), look at random node. If node has average_graph_degree - 1 $<=$ degree $<=$ average_graph_degree + 1 infect that node

Loop

- For all I_a nodes:
  - +1 to days infected. If this results in $>=$ 8 infected days, flip a coin for recovery
    - If succesful, recovered (remove all edges connected to node)
  - For all I_m nodes:
    - +1 to days infected.
      - If this results in started symptoms. Remove all edges except for "home" (simulating home quarantine)
        - AND 20% flip of being diagnosed
      - If $>=$ 10 infected days and showing symptoms. Flip a coin with prob $\frac{1}{20-\text{symptom\_start\_day}}$ for recovery (uniform chance of recovery while symptoms showing)
      - If $>=$ 20 infected days, recovered (remove all edges)
  - For all I_s nodes:
    - +1 to days infected
      - If results in started symptoms, +1 occupied hospital beds and +1 diagnosed. AND hospital isolation (remove all edges)
      - If $>=$ 20 infected days, flip coin for death (15% over 5 days)
        - if dead, -1 occupied beds
      - If $>=$ 25 infected days, recovered and -1 occupied beds
  - For all I_c nodes:
    - +1 to days infected
      - If results in started symptoms, +1 to occupied UTI beds (intensive care), +1 diagnosed. AND hospital isolation (remove all edges)
      - If $>=$ 21 infected days, flip coin for death (50% over 4 days)
        - If head, -1 occupied UTI beds
      - If $>=$ 25 infected days, -1 occupied UTI beds and recovered
  - For all edges between "S" and any infect node:
    - flip a coin according to edge weight
      - if succesful, flip a coin for assignment to infection state [I_a, I_m, I_s, I_c] = [0.3, 0.55, 0.1, 0.05]. And choose day symptoms will start (2-14 with log-normal (Weibull) distribution with mean=6.4 days and stddev=2.3).
        - If assigned to I_s or I_c, after choosing symptom start day set symptom floor at 9 and 8 respectively (can't be lower than this)

- On "action" days (March 24 and May 25)
  - remove layers assigned to day as removed
    - March 24 = transport, school, church
    - May 25 = work
  - replace layers assigned to day as restored
    - Currently none
  - For each removed layer, boost the "home" layer weights by 20% (so on march 24th, removed 3 layers and home layer weights get multiplied by 1.6)