June 17 REU Talk

Simulating antibiotic resistant bacteria populations

Recap

- The goal of our project is to simulate growing bacteria populations in order to explore population dynamics.
- We have two populations – susceptible and resistant. Each grows at its own constant rate, but the resistant population grows slightly more slowly.
- Susceptible can turn into resistant by acquiring a plasmid, called transformation.

This week

Continued work

- Generated some plots of population vs time, population at a given time vs growth rate, and population at a given time vs transformation rate.
- This shows us how changes in growth rates affect the population over time.
- We saw that within reasonable parameter ranges, changes in transformation rate had a much more drastic impact on long term population size than changes in birth rates.
- We also saw some interesting crossover behaviors – some of the parameter combinations that led to the smallest long-term populations had the highest populations early on.
- Solved population equations to determine at what time the susceptible and resistant populations would be equal, and analyzed how changing the parameters affected that.
- This let us change the parameter ranges we're using to try and make the crossover point (and any potentially interesting associated behaviors) occur within the times we're looking at.

Multiprocessing

- I took some time to characterize runtimes of these simulations
- The determining factors in how long an individual simulation will run are population size being simulated, and timestep we're simulating to.
  - The number of timesteps our algorithm performs goes exponentially with max simulation time.
- For a relatively short simulation time (running until T=15), each simulation run took about 10 seconds on my laptop, which is about twice as fast as the linux remote (20 seconds). Running on the cluster is about 3x faster.
  - For reference, interesting crossover behaviors tend to show up between T=5 and T=32, more heavily biased towards longer times.
- So, in order to see these crossover behaviors, we have to run for longer times. But increasing just from T=15 to T=16 doubled the runtime. I did a few different times, and this seemed consistent.
- Large numbers of simulation runs (both for statistical analysis and to cover a large range of parameter values) would take very long
- Multiprocessing - when the program starts up simulation runs, it splits them off into parallel processes. This allows them to work simultaneously but independently of the others.
- Running on the cluster showed a 12x speedup over simulations on my laptop, or a 24x speedup over running on the linux remote.

Next steps

- Want to start doing statistical analysis on simulation runs and characterize fluctuations in populations.
  - How do changes in initial conditions affect population dynamics?
- More lit review
- Implement carrying capacity / logistic growth
  - Upper limit on population size
- Transformation rate dependent on population size (Hill function)