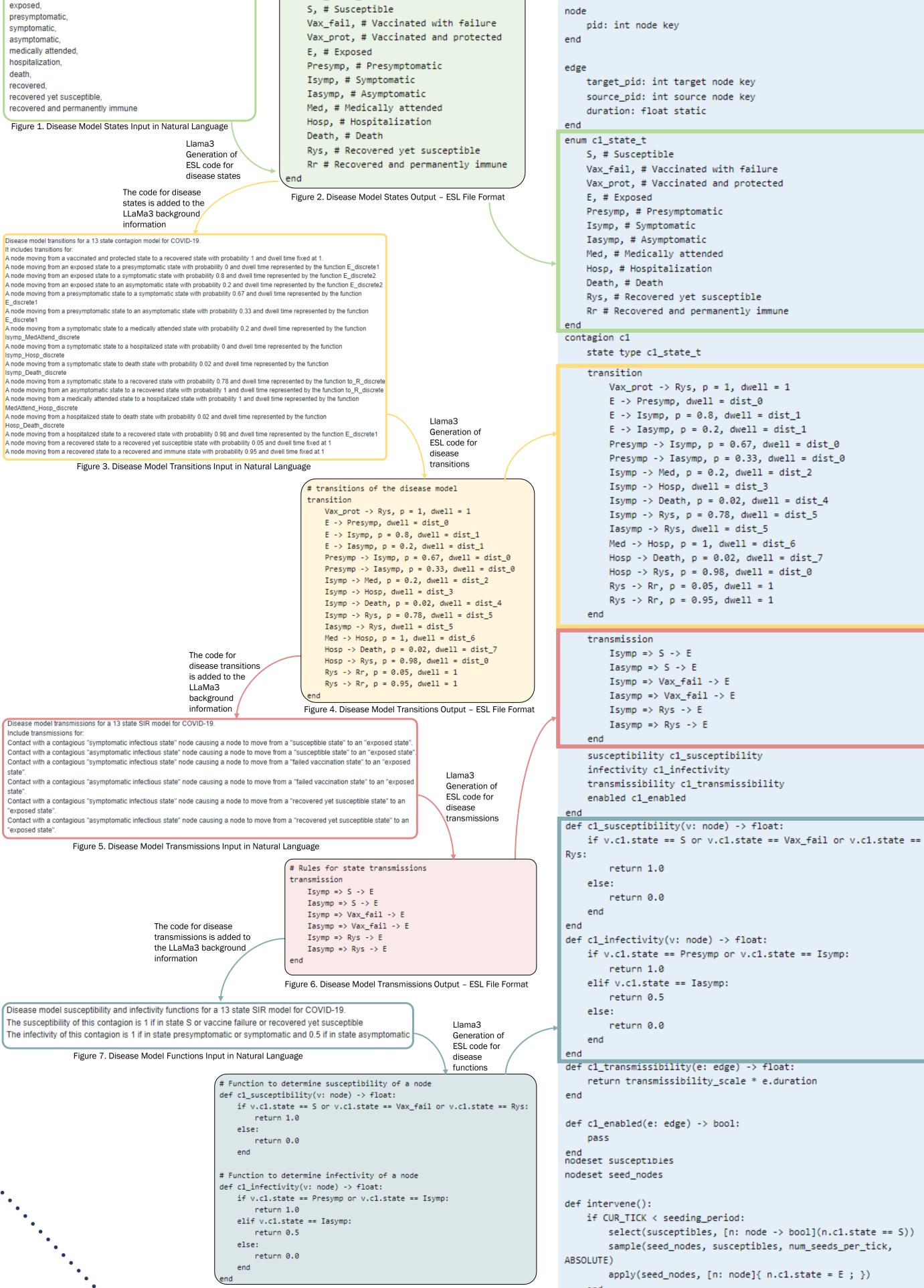
EpiGen: Generating HPC Epidemic Simulators using Natural Language by leveraging Large Language Models

Libby Trainum Mentor: Parantapa Bhattacharya

- Creating efficient high-performance computing (HPC) epidemic simulators can be difficult for researchers
- Large language models (LLM) are good at parsing complex system descriptions from natural language
- Episim37 is framework for writing HPC epidemic simulators that run on top of social contact networks
- EpiGen is a system for creating HPC epidemic simulators using Episim37 and LLMs
- EpiGen takes in natural language descriptions of compartmental disease model and generates an Episim37 simulator using the ESL37 domain-specific language
- Use of natural language reduces technical barriers for researchers for creating efficient epidemic simulations and accelerates the setup phase of epidemiological research

| Simulator Generation Process | | |
|-----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| | <pre># Simple Disease Model for EpiSim37 # Generated by LLaMa from disease model description</pre> | Episim37 |
| ease model states for a 13 state contagion model for COVID-19. cludes states for: cinated with failure, cinated and protected, | <pre>config transmissibility_scale: float = 0.3 config seeding_period: int = 10 config num_seeds_per_tick: int = 10</pre> | High performance simulation framework for modeling epidemics on top of social contact |



networks using custom domain-specific language

- Significant advancement over its predecessor, EpiHiper, with ~30x efficiency and ~15x fewer computational resources (for whole US simulations)
- Allows comprehensive analysis and modeling of disease spread
- Easily adaptable to a wide range of epidemic scenarios and hypotheses
- Powerful tool for researchers and public health officials

ESL37 Domain-Specific Language

- ESL37 is a domain-specific language for specifying epidemic simulations for Episim37
- Describes 2 components, the social contact network and the contagion model
- Each node in the social contact network represents a member of a population
- Edges in the social contact network represent interactions between nodes where disease transmission can occur
- Disease states represent various phases of disease progression
- Transitions define rules for moving from one disease state
- Transmissions define disease spread due to contact between nodes
- Allows for the definition of interventions, like seeding infections or the simulating of public health responses

Figure 6. Disease Model Functions Output - ESL File Format

| nd | |
|-----------|------------------------------------------------------------------------|
| lef | <pre>c1_infectivity(v: node) -> float:</pre> |
| | if v.c1.state == Presymp or v.c1.state == Isymp: |
| | return 1.0 |
| | elif v.c1.state == Iasymp: |
| | return 0.5 |
| | else: |
| | return 0.0 |
| | end |
| nd | |
| ef | <pre>c1_transmissibility(e: edge) -> float:</pre> |
| | return transmissibility_scale * e.duration |
| nd | |
| _ | |
| e† | <pre>c1_enabled(e: edge) -> bool:</pre> |
| | pass |
| nd ode | eset susceptibles |
| ode | eset seed_nodes |
| | |
| ef | intervene(): |
| | if CUR_TICK < seeding_period: |
| | <pre>select(susceptibles, [n: node -> bool](n.c1.state == S))</pre> |
| | <pre>sample(seed_nodes, susceptibles, num_seeds_per_tick,</pre> |
| BSC | DLUTE) |
| | apply(seed nodes, [n: node]{ n.cl.state = E : }) |

-}) priv(seen_nones, [i end

Figure 9. Final Output - ESL37 File for Disease Model Described

Large Language Models

- Large language models leverage deep learning techniques
- For this work, we used LLaMa3 and Gemma
- EpiGen translates natural language descriptions into ESL37 using LLaMa3
- Provides an incremental method to generate the disease simulation
- Simplifies setup and adjustment of epidemic models
- Architecture and training on example files ensure accuracy
- Further training and refinement for broader array of diseases and scenarios is possible--ongoing learning process improves utility and precision over time
- Integration makes simulation file generation efficient, accessible, and responsive

Computing for Global Challenges

UNIVERSITY of VIRGINIA

BIOCOMPLEXITY INSTITUTE