1) Location

1.1) Overview

Class:

- `Location` in `location.py`

It is initialized with a country/region/city from the real world from a .geojon file with its topology. It is initialized with the name of the original location plus an unique identifier (e.g., biobio-4afb) as the configuration of its habitants and their connections are unique.

A location is composed by **sublocations**, which are composed by an identifier, a geometry (the polygon that is plotted as its representation), and a centroid in which the label of the sublocation will be displayed when plotted. It also contains other relevant information from the original region used to construct it but mostly it is not used. This information is stored in the `data` instance variable.

Jupyter notebooks to create locations:

- map_gen_biobio.ipynb
- map_gen_brazil.ipynb (for zacatecas)
- map_gen_ica.ipynb

1.2) Population

Each sublocation of a location is setup with a population of a particular size. So far, there are two ways to select this size:

- Based on the population of the original location, i.e., each sublocation has a different density
- All sub locations have the same population density, which leads to more interesting patterns so far.

Whichever the population size, each individual is positioned on the sublocation by following a homogenous geographic distribution.

1.3) Network

The network of a location indicates how the population is connected in order for a disease to spread. For now, we are using a simple way to compute the network using a geographical threshold network[2] which connect people based on their distance.

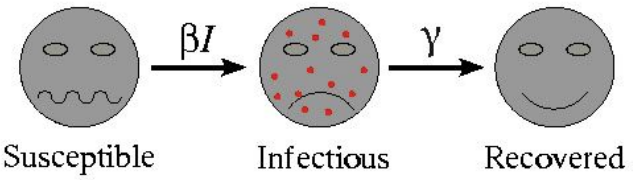
2) Propagation

Classes:

- **Main:** `PropagationSimulator` from `propagationsimulator.py`
- **Model:** `SIRModel` from `sirmodel.py`
- **Process simulator:** `HistoryStochasticDynamics` from `historystochasticdynamics.py` (modification from the original class `StochasticDynamics` that allows me to store the history of each time. I'm still not sure if it's stored somewhere else and I was not able to find it)
- **Animation:** `PropagationCountryAnimator` from `propagationcountryanimator.py`

The propagation of a disease is simulated using Epydemic[3] a framework for scalable and efficient simulation of epidemic processes which uses compartmental models below[4]. It has two main types of compartment models:

- SIR (susceptible, infected, recovered/removed)



Movement rates between classes of the SIR model

- SIS (Susceptible, infected, susceptible)

We use the SIR which is one of the most common used.

The process is simulated with steps, in which step is computed a new event which can be infect or remove an infected person. There are two ways of running the process:

---

[2] https://networkx.github.io/documentation/latest/reference/generated/networkx.generators.geometric.geographical_threshold_graph.html?highlight=geographic#networkx.generators.geometric.geographical_threshold_graph

[3] https://pyepydemic.readthedocs.io/en/latest/index.html

[4] https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology

- Synchronous: each step is computer by increasing the time in discrete intervals (i.e., in each cycle t = t +1)
- Stochastic: in which time is continuous and the difference between two events does not have a fixed interval.

We use Stochastic as it leads to more realistics results.

2.1) Parameters

Epydemic defines 2 main probabilities to start a simulation:

- pInfect -> the probability of infecting another person given
- pRemove -> the probability that an infected node is recovered/removed

It also allows a third parameter:

- pInfected -> the probability a person is initially infected

For now we are not using the third parameter as we manually select a sublocation in which we take a random sample that will be the seeds of the disease.

2.2) Seeds

A disease starts with a set of infected people within a location which are called seeds. A seeds configuration is initialized by the location id + its unique id (e.g., biobio-4afb_seeds_d0fc)

2.3) Vaccination

Vaccinated people are not susceptible to get the disease. They allow us to make walls among sublocations that will stop the propagation of a disease. For now, vaccination is applied to a complete sublocation.

2.4) Jupyter notebooks:

- 9_create_training_dataset.ipynb
- 10_create_L1_biobio_datasets.ipynb
- 10_create_L2_zacatecas_datasets.ipynb