Epigrass Documentation

Release 2.0a1

Flavio Codeco Coelho

July 07, 2008

CONTENTS

Contents:

CHAPTER

ONE

Overview of Epigrass

Epigrass is a platform for network epidemiological simulation and analysis. It enables researchers to perform comprehensive spatio-temporal simulations incorporating epidemiological data and models for disease transmission and control in order to create sophisticated scenario analyses.

1.1 Components

The EpiGrass system is composed of 4 components, The EpiGrass simulator, the EpiGrass database, EpiGrass visualization module, and the report generator. They can all be used separately but should be invoked through EpiGrass' user interface.

1.2 Modeling Approach

The geographical networks over which epidemiological processes take place can be very straightforwardly represented in a object-oriented framework. In such a framework, the nodes and edges of the geographical networks are objects with their attributes and methods.

Once the archetypal node and edge objects are defined with appropriate attributes and methods, then a code representation of the real system can be constructed, where cities (or other geographical localities) and transportation routes are instances of the node and edge objects, respectively. The whole network is also an object with a whole collection of attributes and methods.

This framework leads to a compact and hierarchical computational modelconsisting of a network object containing a variable number of node and edge objects. This framework also do not pose limitations to encapsulation, potentially allowing for networks within networks if desirable (not yet implemented).

For the end user this framework is transparent since it mimics the natural structure of the real system. Even after the model is converted into a code object all of its component objects remain accessible to the user.

1.3 Geographical Network Models

EpiGrass's geo-referenced models are built from two basic sources of data: GRASSindex{GRASS} vector layerindex{GRASS!vector layer} which provide the cartographical base over which the models are specified and specific data about nodes and edges that are provided by the user for the network of interest.

1.3.1 Defining the Cartographic Background

If the user has a GRASS locationindex{GRASS!location} for the georeferred data, this can be passed to EpiGrass. In this case, the cartographic background is defined by selecting a GRASS Location datasetfootnote{See GRASS documentation on how to setup a Location}. From this Location various types information can be extracted for inclusion in the model's definition of visualization.

If the user does not have a GRASS location, he can still use EpiGrass. In this case, the georeferred data is read only from two .csv files (more on that ahead).

1.3.2 Defining Nodes

A graph has nodes and edges. Nodes can be cities or other localities depending on the model being constructed. The definition of nodes require the setting of many attributes listed below. The nodes will have many more attributes defined at run-time which will depend on other aspects of the model, these will be discussed later.

The data necessary at build time to create nodes come from a CSV (comma-separated-values) ASCII-text file, with the following attributes, (in this order):

Latitude, Longitude This attribute will be used to geo reference the node. The coordinate system must match those used in the cartographic base imported from GRASS. Coordinates can be coded in either decimal or sexagesimal format.

Name Used for identification purposes only. It can be a city name, for instance.

Population Since the simulation models will all be of a populational nature. Population size must be specified at build time.

Geocode A Unique Geocode (a number) is required. It will be used as a label/index to facilitate reference to specific nodes.

1.3.3 Defining Edges

In EpiGrass' graphs, edges represent transportation routes. Similarly to nodes, edges are defined at build-time with a reduced set of attributes which will be extended at run-time. EpiGrass also expects to get these attributes from a CSV file:

Source The name of the source node. The edges are bi-directional, but the nodes are labeled source and destination for reference purposes.

Destination The name of the destination node.

Forward migration Migration rate from source to destination, in number of persons per unit of time.

Backward migration Migration rate from destination to source, in number of persons per unit of time.

Length Distance in kilometers (or another unit) from source to destination via the particular route (not straight line distance).

Source's geocode This is the unique numerical identifier used in the sites file.

Destination's geocode This is the unique numerical identifier used in the sites file.

1.3.4 Defining models

The word model in EpiGrass can mean two distinct objects: The network model and the node's epidemic model.

Node objects, in an EpiGrass model, contain well-mixed population dynamic models within them. These models determine the dynamics of epidemics within the particular environments of each node. EpiGrass comes with a few standard epidemiological modelsindex{Models!epidemiological models} to choose from when setting up your network. Currently, The same model type is applied to every node although their parameterization is node-specific. Besides the built-in model types, users can define their own, as shown on chapter ref{ch:usingepg}.

Network modelsindex{Models!network model} are specified in a ASCII-text script file (see appendix ref{script}). EpiGrass comes with a few demo Network models for the user to play with until he/she is confident enough to build their own. Even then, it is advisable to use the demo scripts provided as templates to minimize syntax errors.

The script on the appendix ref{script} specifies a network model with an stochastic SEIR (see ref{cap:modeling}) epidemic model in its nodes. The user should study this model and play with its parameters to understand the features of EpiGrass. A step-by-step tutorial on how to edit the model script can be found on chapter ref{ch:usingepg}.

1.4 The Simulation

A simulation run in EpiGrass consists of a series of tasks performed at each time step [1] .

Calculate migration For all edges in the network, the number of persons traveling each way is determined for the current time-step.

Run epidemic models For each node in the network the epidemic demographics are updated based on the local number of infected and susceptible individuals which have been updated by the transportation system.

All aspects of the simulation such as number of passengers traveling on each edge, number of infected/susceptible on each node and etc., are recorded in a step-by-step basis. This complete record allows for the model to be analyzed after the simulation has been completed without having to recalculate it.

1.4.1 Output

The output of a simulation in EpiGrass is three-fold: A graphical display which the animated outcome of the simulation, a written report, and a database table with numeric results.

Graphical display

During a simulation, selected epidemiological variables are animated in a 3-dimensional rendering over the map of the region containing the network.

Report Generation

The report contains a detailed analysis of the network model and the simulations ran with it. The report generates a LaTeX source file and compiles it to a PDF document for visualization.

Three types of report are currently available:

Report = 1 Returns a set of descriptors of the network, described in chapter

Report = 2 Returns a set of basic epidemiological measures and plots of the time series.

Report = 3 Report 1 + Report 2

Report Generation is an optional, though recommended, step done at the end of the simulation. For the report, descriptive statistics are generated for the network. These have to do with network topology and properties. Additional sections can be added to the report with basic statistical analyses of the output of pre-selected nodes [2] .

Database output

Time series of L,S,E,and I, from simulations, are stored in a MySQL database named *epigrass* index{Database!epigrass database}. The results of each individual simulation is stored in a different table named after the model's script name, the date and time the simulation has been run. For instance, suppose you run a simulation of a model stored in a file named texttt{script.epg}, then at the end of the simulation, a new table in the epigrass database will be created with the following name: texttt{script_Wed_Jan_26_154411_2005}. Thus, the results of multiple runs from the same model get stored independently.index{Database!results table}

[1]The number of time steps is defined in the model script

[2]Listed in the siteRep variable at the script

CHAPTER

TWO

Indices and tables

- *Index*
- *Module Index*
- *Search Page*