Stride project

User Manual

Version 1.0 (June 18, 2018) .

Centre for Health Economics Research & Modeling of Infectious Diseases, Vaccine and Infectious Disease Institute, University of Antwerp.

Modeling of Systems and Internet Communication, Department of Mathematics and Computer Science, University of Antwerp.

Interuniversity Institute for Biostatistics and statistical Bioinformatics, Hasselt University.

Willem L, Kuylen E & Broeckhove J

Contents

CHAPTER 1

Introduction

This manual provides a brief description of the Stride software and its features. Stride stands for **S**imulation of **tr**ansmission of **i**nfectious **dis**e**a**ses and is an agent-based modeling system for close-contact disease transmission developed by researchers at the University of Antwerp and Hasselt University, Belgium. The simulator uses census-based synthetic populations that capture the demographic and geographic distributions, as well as detailed social networks. Stride is an open source software. The authors hope to make large-scale agent-based epidemic models more useful to the community. More info on the project and results obtained with the software can be found in: *"Willem L, Stijven S, Tijskens E, Beutels P, Hens N & Broeckhove J. (2015) Optimizing agent-based transmission models for infectious diseases, BMC Bioinformatics, 16:183"* [1].

The model population consists of households, schools, workplaces and communities, which represent a group of people we define as a "cluster". Social contacts can only happen within a cluster. When school or work is off, people stay at home and in their primary community and can have social contacts with the other members. During other days, people are present in their household, secundary community and a possible workplace or school.

We use a *Simulator* class to organize the activities from the people in an *Area*. The Area class has a *Population*, different *ContactPool* objects and a *ContactHandler*. The *ContactHandler* performs Bernoulli trials to decide whether a contact between an infectious and susceptible person leads to disease transmission. People transit through Susceptible-Exposed-Infected-Recovered states, similar to an influenza-like disease. Each *ContactPool* contains a link to its members and the *Population* stores all personal data, with *Person* objects. The implementation is based on the open source model from Grefenstette et al. [2]. The household, workplace and school clusters are handled separately from the community clusters, which are used to model general community contacts. The *Population* is a collection of *Person*

objects.

CHAPTER 2

Software

2.1 System Requirements

Stride is written in C++ and portable over all platforms that have the GNU C++ compiler. To use the visualalization tools Qt is needed (at least version 5.9). The following tools needs to be installed:

- g++
- make
- CMake (¿= v3.10)
- Boost
- Python (optional, for automatization)
- Doxygen (optional, for documentation)
- LaTeX (optional, for documentation)

2.2 Installation

To install the project, first obtain the source code by cloning the repository to a directory or download a zip file with all project material from the GitHub website and de-compress the archive. The build system for Stride uses the CMake tool. This is used to build and install the software at a high level of abstraction and almost platform independent (see http://www.cmake.org/). The project includes

the conventional make targets to "build", "install", "test" and "clean" the project. There is one additional target "configure" to set up the CMake/make structure that will actually do all the work. For those users that do not have a working knowledge of CMake, a front end Makefile has been provided that invokes the appropriate CMake commands. More details on building the software can be found in "INSTALL.txt" in the source folder.

2.3 Documentation

The Application Programmer Interface (API) documentation is generated automatically using the Doxygen tool ((see `www.doxygen.org`) from documentation instructions embedded in the code .

The user manual distributed with the source code has been written in LaTeX(see `www.latex-project.org`).

2.4 Directory layout

The project directory structure is very systematic. Everything used to build the software is stored in the directory `./src`:

- `src/main`: Code related files (sources, third party libraries and headers, ...)
  - `src/main/<language>`: source code, per coding language: cpp (for C++), python, R,
  - `src/main/resources`: third party resources included in the project:
- `src/doc`: documentation files (API, manual, ...)
  - `src/doc/doxygen_ref_man`: files needed to generate the reference documentation with Doxygen
  - `src/doc/latex_man`: files needed to generate the user manual with Latex
- `src/test`: test related files (scripts, regression files, ...)

2.5 File formats

The Stride software supports different file formats:

CSV
Comma separated values, used for population input data and simulator output.

HDF5
Hierarchical Data Format 5, designed to store and organize large amounts of data.

JSON
JavaScript Object Notation, an open standard format that uses human-readable text to transmit objects consisting of attribute-value pairs. (see www.json.org)

TXT
Text files, for the logger.

XML
Extensible Markup Language, a markup language (both human-readable and machine-readable) that defines a set of rules for encoding documents.

Proto
Protocol Buffers, used for exporting and importing the generated population and geographical grid.

2.6 Testing

Unit tests and install checks are added to Stride based on Google's "gtest" framework and CMake's "ctest" tool. In addition, the code base contains assertions to verify the simulator logic. They are activated when the application is built in debug mode and can be used to catch errors at run time.

2.7 Results

The software can generates different output files:

cases.csv
Cumulative number of cases per day.

summary.csv
Aggregated results on the number of cases, configuration details and timings.

person.csv
Individual details on infection characteristics.

logfile.txt
Details on transmission and/or social contacts events.

gengeopop.proto
Generated population and geographical grid.

map.png
Image of the map shown in the MapViewer component.

calibration_data.json
Results of the calibration component.

CHAPTER 3

Simulator

3.1 Workspace

By default, Stride is installed in `./target/installed/` inside de project directory though this can be modified using the CMakeLocalConfig.txt file (example is given in `./src/main/resources/make`). Compilation and installation of the software will create the following files and directories:

- Binaries in directory `<project_dir>/bin`
  - *stride*: executable.
  - *gtester*: regression tests for the sequential code.
  - *gengeopop*: generates the population and geographical grid.
  - *guilauncher*: tool to edit a simulation configuration and execute it
  - *mapviewer*: tool to visualize the population and geographical grid, can be directly used from *stride* and *guilauncher*.
  - *wrapper_sim.py*: Python simulation wrapper
- Configuration files (xml and json) in directory `<project_dir>/config`
  - $run\_default.xml$: default configuration file for Stride to perform a Nassau simulation.
  - $run\_generate\_default.xml$: default configuration file for Stride to first generate a population and geographical grid and then perform a Nassau Simulation.
  - $run\_import\_default.xml$: default configuration file for Stride to first import a population and geographical grid and then perform a Nassau Simulation.

  - *run_regions_default.xml*: default configuration file for Stride to first generate several regions with their own population and geographical grid and then perform a Nassau Simulation.
  - *run_regions_import.xml*: default configuration file for Stride to first import several regions with their own population and geographical grid and then perform a Nassau Simulation.
  - *run_miami_weekend.xml*: configuration file for Stride to perform Miami simulations with uniform social contact rates in the community clusters.
  - *wrapper_miami.json*: default configuration file for the wrapper_sim binary to perform Miami simulations with different attack rates.
  - ...

- Data files (csv) in directory `<project_dir>/data`
  - *belgium_commuting*: Belgian commuting data for the active populations. The fraction of residents from "city_depart" that are employed in "city_arrival". Details are provided for all cities and for 13 major cities.
  - *belgium_population*: Relative Belgian population per city. Details are provided for all cities and for 13 major cities.
  - *contact_matrix_average*: Social contact rates, given the cluster type. Community clusters have average (week/weekend) rates.
  - *contact_matrix_week*: Social contact rates, given the cluster type. Community clusters have week rates.
  - *contact_matrix_week*: Social contact rates, given the cluster type. Primary Community cluster has weekend rates, Secondary Community has week rates.
  - *disease_xxx*: Disease characteristics (incubation and infectious period) for xxx.
  - *holidays_xxx*: Holiday characteristics for xxx.
  - *pop_xxx*: Synthetic population data extracted from the 2010 U.S. Synthetic Population Database (Version 1) from RTI International for xxx [3, 4].
  - *ref_2011*: Reference data from EUROSTAT on the Belgian population of 2011. Population ages and household sizes.
  - *ref_fl2010_xxx*: Reference data on social contacts for Belgium, 2011.
  - *DE_cities.csv*: German cities, fetched from [1].
  - *FR_cities.csv*: French cities, fetched from [2].
  - *NL_cities.csv*: French cities, fetched from [3].
  - *wallonia_cities.csv*: Wallonia cities, fetched from [4].
  - *submunicipalities.csv*: Belgian sub-municipalities, fetched from [5].

1
2
3
4
5

- Documentation files in directory `./target/installed/doc`
  - Reference manual
  - User manual

3.2 Run the simulator

From the workspace directory, the simulator can be started with default configuration using the command "`./bin/stride`". Settings can be passed to the simulator using one or more command line arguments:

- `-c` or `--config`: The configuration file.
- `-r` or `--r0`: To obtain the basic reproduction number, no tertiary infections.

3.3 Generating a population and geographical grid

From the workspace directory, the generation of a population and geographical grid (sometimes called GeoGrid) can be started with the default configuration using the command "`./bin/gengeopop`". The following configuration options are available:

`--populationSize`
: The size of the population to generate. By default a population of `6000000` is generated.

`--fracActive`
: The fraction of people who are active, i.e. who are employed or students. By default 0.75 is used.

`--fracStudentCommuting`
: The fraction of students commuting. By default `0.5` is used.

`--fracActiveCommuting`
: The fraction of active people who commutes. By default 0.5 is used.

`--frac1826Students`
: The fraction of 1826 years which are students. By default 0.5 is used.

`--household`
: The file to read the household profiles from.

`--commuting`
: The file to read the commuting information from.

`--cities`
: The file to read the cities from.

`--subMinicipalities`
: The file to read the sub-municipalities from.

`--output`
: The file to write the GeoGrid to. By default this is `gengeopop.proto`

`--state`
: The state to be used for initializing the random engine. This can be used to continue with the same state when generating multiple regions.

`--rng_type`
: The type of random engine to use by default `mrg2` is used. Available options are: `mrg2`, mrg3, yarn2, `yarn3`, `lgc64` and `lgc64_shift`.

`--seed`
: The seed to be used for the random engine. The default is `0`.

`--loglevel`
: The loglevel to use, by default this is `info`.

3.4 Using the MapViewer

The MapViewer component can be used to explore a generated GeoGrid. From the workspace directory, the MapViewer can be started by using "`./bin/mapviewer`". No additional command line options are available. To open a GeoGrid use the menu options `File -> Open`. After loading the file, the viewport can be changed so that all markers are visible (`View -> Fit viewport`). The circular markers indicate a city or sub-municipality. The square markers indicate the "parent" city of sub-municipalities, they don't have a population. By clicking on the square marker, all the sub-municipalities will be selected. When one ore more cities is selected the details are shown in the right sidebar. For example the ContactCenters are listed. The ContactPools of a ContactCenter can be shown by clicking on a ContactCenter.

Multiple cities can be selected by holding the control key and left mouse button and selecting a rectangle.

The visible area of a map can be exported to an image file using `File -> Export to image`.

3.5 Python Wrapper

A Python wrapper is provided to perform multiple runs with the C++ executable. The wrapper is designed to be used with .json configuration files and examples are provided with the source code. For example:

```
./bin/wrapper_sim --config ./config/wrapper_default.json
```

will start the simulator with each configuration in the file. It is important to note the input notation: values given inside brackets can be extended (e.g., “rng_seeds”=[1,2,3]) but single values can only be replaced by one other value (e.g., “days”: 100).

Bibliography

[1] L. Willem, S. Stijven, E. Tijskens, P. Beutels, N. Hens, and J. Broeckhove, "Optimizing agent-based transmission models for infectious diseases.," *BMC Bioinformatics*, vol. 16, p. 183, 2015.

[2] J. J. Grefenstette, S. T. Brown, R. Rosenfeld, J. DePasse, N. T. Stone, P. C. Cooley, W. D. Wheaton, A. Fyshe, D. D. Galloway, A. Sriram, H. Guclu, T. Abraham, and D. S. Burke, "FRED (A Framework for Reconstructing Epidemic Dynamics): an open-source software system for modeling infectious diseases and control strategies using census-based populations," *BMC public health*, vol. 13, no. 1, p. 940, 2013.

[3] RTI International, "2010 RTI U.S. synthetic population ver. 1.0," *Downloaded from internet URL: http://www.epimodels.org/midas/pubsyntdata1.do*, 2014.

[4] W. Wheaton, "2010 U.S. synthetic population quick start guide. RTI international," *Retrieved from http://www.epimodels.org/midasdocs/SynthPop/2010_synth_pop_ver1_quickstart.pdf*, 2014.