emodpy-generic

Institute for Disease Modeling

Apr 21, 2022

CONTENTS

emodpy-generic is a collection of Python scripts and utilities created to streamline user interactions with EMOD and idmtools for modeling generic diseases. Much of the functionality is inherited from the emod_api package and emodpy package.

Additional information about how to use idmtools can be found at in Welcome to idmtools. Additional information about EMOD HIV parameters can be found in EMOD parameter reference.

If you have questions, see Frequently asked questions for general questions about all emodpy functionality or *Frequently asked questions* for functionality specific to emodpy-generic.

See Welcome to idmtools for a diagram showing how idmtools and each of the related packages are used in an end-to-end workflow using EMOD as the disease transmission model.

CHAPTER

ONE

EMODPY-GENERIC INSTALLATION

Follow the steps below to install emodpy-generic.

1.1 Prerequisites

First, ensure the following prerequisites are met.

- Windows 10 Pro or Enterprise, Linux, or Mac
- Python 3.9 64-bit (https://www.python.org/downloads/release)
- A file that indicates the pip index-url:
  - Windows
  - Linux

  In C:\Users\Username\pip\pip.ini, containing the following:

```
[global]
index-url = https://packages.idmod.org/api/pypi/pypi-production/simple
```

  In $HOME/.config/pip/pip.conf, containing the following:

```
[global]
index-url = https://packages.idmod.org/api/pypi/pypi-production/simple
```

1.2 Installation instructions

1. Open a command prompt and create a virtual environment in any directory you choose. The command below names the environment "v-emodpy-generic", but you may use any desired name:

```
python -m venv v-emodpy-generic
```

2. Activate the virtual environment:
   - Windows
   - Linux

   Enter the following:

```
v-emodpy-generic\Scripts\activate
```

Enter the following:

```
source v-emodpy-generic/bin/activate
```

3. Install emodpy-generic packages:

```
pip install emodpy-generic
```

If you are on Linux, also run:

```
pip install keyrings.alt
```

4. When you are finished, deactivate the virtual environment by entering the following at a command prompt:

```
deactivate
```

CHAPTER

TWO

API REFERENCE

2.1 emodpy_generic package

The emodpy-generic module is intended to house scripts and tools that enable disease modelers to work more easily with the IDM EMOD Generic model.

2.1.1 Subpackages

emodpy_generic.demographics package

Submodules

emodpy_generic.demographics.GenericDemographics module

class emodpy_generic.demographics.GenericDemographics.**GenericDemographics**(*nodes*, *idref='Gridded world grump2.5arcmin'*, *base_file=None*)

Bases: emod_api.demographics.Demographics.Demographics

This class is derived from emod_api.demographics' Demographics class so that we can set certain defaults for Generic in construction. As we add other disease types, the generalizations and speicfics will become clearer.

emodpy_generic.demographics.GenericDemographics.**fromBasicNode**(*lat=0*, *lon=0*, *pop=1000000.0*, *name=1*, *forced_id=1*)

This function creates a single-node GenericDemographics instance from the params you give it.

Parameters

- **lat** – latitude (not really used)
- **lon** – longitude (not really used)
- **pop** – population. Defaults to 1 million.
- **name** – node name (not really used)
- **forced_id** – node id (not really used)

Returns GENERIC_SIM demographics instance which can be customized and/or written to file.

emodpy_generic.interventions package

The below imports let you do stuff like: from emodpy_generic.interventions import * a = SimpleSIA()

If you want to access all the iv function creates directly without scoping/qualifying them.

Submodules

emodpy_generic.interventions.complex_import module

emodpy_generic.interventions.complex_import.**ComplexImportationEvent**(*camp*, *dips=[0.00012919896640826872*, *durs=[100000]*, *timestep=1*, *nods=[]*)

Seed infection (over time) in the modeled population.

Be careful when configuring import pressure in multi-node simulations. Daily_Import_Pressures defines a rate of per-day importation for each node that the intervention is distributed to. In a 10 node simulation with Daily_Import_Pressures = [0.1, 5.0], the total importation rate summed over all nodes will be 1/day and 50/day during the two time periods. You must divide the per-day importation rates by the number of nodes. Note that there is no control over the ages of the individuals imported, or their monte-carlo weight, like you might find with the Outbreak intervention.

Parameters

- **dips** – DailyImportPressures. An array of rates of per-day importation (per-node).
- **durs** – An array of durations over which to apply import pressure. Goes with dips.
- **timestep** – WHEN: Day to start campaign event.
- **nods** – WHERE: List of nodes at which infections will be seeded.

Returns Campaign event that can be added to campaign.

emodpy_generic.interventions.emergence module

emodpy_generic.interventions.emergence.**basicEmergenceEvent**(*camp*, *genome=0*, *clade=0*, *start_day=1*, *nods=[]*)

Seed infection at a particular time by importing a new infected individual.

Note that this function imports 20 individuals age 5. This could be made configurable but is not yet because this was created with a particular use case in mind. All other values are set to schema defaults.

Parameters

- **genome** – Genome of new infection.
- **clade** – Clade of new infection.
- **start_day** – WHEN: Day to start campaign event.
- **nods** – WHERE: List of nodes at which infections will be seeded.

Returns Campaign event that can be added to campaign.

emodpy_generic.interventions.emergence.**basicSIAEvent**(*camp*, *cov=1.0*, *genome=0*, *clade=0*, *start_day=1*, *nods=[]*)

Seed infection at a particular time by infected existing individuals (or agents really) in the simulation.

Note that the targeted ages are 0-5 years, not configurable without code change.

Parameters

- **clade** – Clade of new infection.
- **genome** – Genome of new infection.
- **cov** – WHO. Percentage of population to infect (given other targeting constrants).
- **start_day** – WHEN: Day to start campaign event.
- **nods** – WHERE: List of nodes at which infections will be seeded.

Returns Campaign event that can be added to campaign.

emodpy_generic.interventions.emergence.**new_intervention_as_file**(*timestep*, *filename=None*)

This function mostly exists for testing so one can exercise the functionality in a single function call.

emodpy_generic.interventions.polio_vaccine module

emodpy_generic.interventions.polio_vaccine.**SimpleIPV1**(*camp*, *timestep=1*, *coverage=1.0*)

IPV (round 1) Campaign Event

AgeAtVaccIInDays is hard-coded to 270.

Parameters

- **coverage** – WHO?
- **timestep** – WHEN?

Returns Campaign event that can be added to campaign.

emodpy_generic.interventions.polio_vaccine.**SimpleIPV2**(*camp*, *timestep=1*, *coverage=1.0*)

IPV (round 1) Campaign Event

AgeAtVaccIInDays is hard-coded to 270.

Parameters

- **coverage** – WHO?
- **timestep** – WHEN?

Returns Campaign event that can be added to campaign.

emodpy_generic.interventions.polio_vaccine.**new_intervention_as_file**(*timestep*, *filename=None*)

This function mostly exists for testing so one can exercise the functionality in a single function call.

emodpy_generic.interventions.simple_sia module

emodpy_generic.interventions.simple_sia.**SimpleSIA**(*camp*, *Event_Name='SIA'*, *timestep=1*, *Target_Age_Min=0.75*, *Target_Age_Max=5.0*, *Coverage=1.0*, *nods=[]*)

Create an intervention that represents a highly-effective acquisition-blocking vaccination.

Parameters

- **Event_Name** –
- **timestep** – WHEN to distribute the intervention.
- **Target_Age_Min** – WHO to distribute the intervention to (lower age bound).
- **Target_Age_Max** – WHO to distribute the intervention to (upper age bound).
- **Coverage** – WHO to distribute the intervention to (%age of population).
- **nods** – WHERE to distribute (list of nodes).

Returns Campaign event that can be added to campaign.

emodpy_generic.interventions.simple_sia.**new_intervention_as_file**(*timestep*, *filename=None*)

This function mostly exists for testing so one can exercise the functionality in a single function call.

CHAPTER

THREE

FREQUENTLY ASKED QUESTIONS

As you get started with emodpy-generic, you may have questions. The most common questions will be added below. For questions related to functionality in related packages, see the following documentation:

- Frequently asked questions for EMOD
- Frequently asked questions for idmtools
- Frequently asked questions for emod-api
- Frequently asked questions for emodpy

PYTHON MODULE INDEX

e

INDEX