emodpy-measles

Institute for Disease Modeling

Apr 17, 2023

CONTENTS

emodpy-measles is a collection of Python scripts and utilities created to streamline user interactions with EMOD and idmtools for modeling measles. 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 generic parameters for modeling measles can be found in EMOD parameter reference.

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.

CONTENTS

1.1 emodpy-measles installation

Follow the steps below to install emodpy-measles.

1.1.1 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
  ```

1.1.2 Standard 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-measles", but you may use any desired name:

   ```
   python -m venv v-emodpy-measles
   ```

2. Activate the virtual environment:

   - Windows

   - Linux

   Enter the following:

```
v-emodpy-measles\Scripts\activate
```

Enter the following:

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

3. Install emodpy-measles packages:

```
pip install emodpy-measles
```

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
```

1.1.3 Developer installation

Use a virtual environment as described in the standard installation. You then have several options if you are modifying the code locally:

Option 1: Create a hard link to package_setup.py available to pip:

```
ln package_setup.py setup.py
```

and then install in the *emodpy-measles* directory:

```
pip install -e .
```

Option 2: Each time you make a change you want to use:

```
python package_setup.py bdist_wheel
pip install dist\emodpy_measles-XXX.whl
```

1.1.4 Windows

To properly install Shapely on Windows (necessary for the emod_api package) and/or if Snappy compression support is desired or needed, consider downloading and installing the latest python-snappy package for Windows from Christoph Gohlke's python package website.

1.1.5 Example

For instructions to run a measles example scenario, see https://github.com/InstituteforDiseaseModeling/emodpy-measles/tree/master/examples/katherine.

1.2 FAQ

As you get started with emodpy-measles, you may have questions. Common questions are answered 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

Contents

- *Why does the system download a new Eradication binary (and supporting binaries/schema) each time I run?*
- *I want to just put a binary (and corresponding schema) in the downloads directory and not doing any more downloads.*
- *What is the purpose of the manifest.py file?*
- *How do I set config parameters?*
- *Are there defaults?*
- *I pip installed emodpy-measles, but I want to make changes. How should I do that?*
- *I notice that I can import emod_api.campaign and use that as an object. I haven't seen that before.*
- *I want to just load a demographics.json, not create one programmatically.*
- *What happens if I don't VPN in?*
- *How do I specify the number of cores?*
- *How do I specify the log level for DTK? I get a schema error when I try to set it now.*
- *How do I do campaign sweeps?*
- *How do I do demographics sweeps?*

1.2.1 Why does the system download a new Eradication binary (and supporting binaries/schema) each time I run?

The system is designed very much like you browse the web. When you go to a website/page, it downloads html, png and other files. If you go there again, it does it again. We don't even think about it (unless we have network issues). We want this to be the New Normal. You get the latest binary from measles-Ongoing branch that Passes All The Tests. That said, we know there are times when stasis and stability are paramount. To that end, you can pass a bamboo build number to 'get_model_files(plan, manifest)' and you'll always get that one.

1.2.2 I want to just put a binary (and corresponding schema) in the downloads directory and not doing any more downloads.

Comment out the call to 'get_model_files'.

1.2.3 What is the purpose of the manifest.py file?

manifest.py is designed to house ALL your input and output paths in a single location. It also includes the path of your choice for where model binaries (and schema) are downloaded to and uploaded from. This is because even though you can happily ignore these files if you want, you'll have a better EMOD experience if you can easily reference the schema file and sometimes it's nice to be able to have access to the binary itself.

1.2.4 How do I set config parameters?

Provide a param-setting function and pass that function to the emodpy task creator (example). In that function, you can set parameters directly (example). Or you can call a function in a standalone script that does the config params (example).

1.2.5 Are there defaults?

Great question. If you didn't set any config params at all, they will have defaults based on the schema. But you can add, for example, a function in emodpy_measles.config called set_team_defaults().

1.2.6 I pip installed emodpy-measles, but I want to make changes. How should I do that?

`python package_setup.py develop` There are some other options but that seems to be the most popular and proven. This means that the emodpy-measles module in site-packages actually points to the same code as you have in your git checked-out repo. This link is useful for details. But we want you to get the changes you like into the tested, versioned, released module before too long.

1.2.7 I notice that I can import emod_api.campaign and use that as an object. I haven't seen that before.

Sure. Python modules are a lot like singletons. There's no need to add a static class inside that module in many cases. Think of the module (which can have variables and methods) as a static class.

1.2.8 I want to just load a demographics.json, not create one programmatically.

OK, but be aware that one of the benefits of emodpy/emodapi is that you get guaranteed consistency between demographics settings and config params. But if you really want to use a raw demographics.json that you are very confident in, you can open that in your demog builder. An example of that is:

```python
def build_demog():
import emod_api.demographics.Demographics as Demographics
demog = Demographics.from_file( "demographics.json" )
return demog
```

1.2.9 What happens if I don't VPN in?

You can't access Bamboo and download the Eradication binaries (including plugins and schema). An alternative system is under development and available in beta to folks who need it.

1.2.10 How do I specify the number of cores?

num_cores is an undocumented param to Platform(). It is not a DTK config param.

1.2.11 How do I specify the log level for DTK? I get a schema error when I try to set it now.

An example of how to set non-schema config params is coming soon.

1.2.12 How do I do campaign sweeps?

You can look at (this example) in Malaria. A campaign sweep example will be added to emodpy-measles.

1.2.13 How do I do demographics sweeps?

A demographic sweep example will be added to emodpy-measles.

1.3 API reference

1.3.1 emodpy_measles package

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

Subpackages

emodpy_measles.demographics package

Submodules

emodpy_measles.demographics.MeaslesDemographics module

class emodpy_measles.demographics.MeaslesDemographics.**MeaslesDemographics**(*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 Measles in construction. Keen observers will note thatt SetDefaultProperties does not look like a measles-specific function. But as we add other disease types the generatlizations and speicfics will become clearer. The architectural point is solid.

emodpy_measles.demographics.MeaslesDemographics.**from_template_node**(*lat=0, lon=0, pop=1000000.0, name=1, forced_id=1*)

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

emodpy_measles.demographics.MeaslesDemographics.**from_params**(*tot_pop=1000000.0, num_nodes=100, frac_rural=0.3, id_ref='from_params'*)

Create a multi-node *MeaslesDemographics* instance as a synthetic population based on a few parameters.

> **Parameters**
>
> - **tot_pop** – The total human population in the node.
>
> - **num_nodes** – The number of nodes to create.
>
> - **frac_rural** – The fraction of the population that is rural.
>
> - **id_ref** – Method describing how the latitude and longitude values are created for each of the nodes in a simulation. "Gridded world" values use a grid overlaid across the globe at some arcsec resolution. You may also generate the grid using another tool or coordinate system. For more information, see Metadata.
>
> **Returns** A `Measles` instance.

emodpy_measles.demographics.MeaslesDemographics.**from_pop_csv**(*pop_filename_in, site='No_Site', min_node_pop=0*)

Create a multi-node `MeaslesDemographics` instance from a CSV file describing a population.

> **Parameters**

- **pop_filename_in** – The path to the demographics file to ingest.

- **pop_filename_out** – The path to the file to output.

- **site** – A string to identify the country, village, or trial site.

Returns A `MalariaDemographics` instance.

emodpy_measles.demographics.MeaslesDemographics.**get_file_from_http**(*url*)
    Get data files from simple http server.

emodpy_measles.interventions package

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

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

Submodules

emodpy_measles.interventions.complex_import module

emodpy_measles.interventions.complex_import.**ComplexImportationEvent**(*camp*,
                                                                     dips=[0.00012919896640826872],
                                                                     durs=[100000],
                                                                     timestep=1, *nodes=None*)

    Import the Disease

    Parameters

    - **DailyImportPressures** – How Much? A rate of per-day importation for each node that
      the intervention is distributed to.

    - **Durations** – How Long? The durations over which to apply import pressure.

    - **timestep** – When? Day to start campaign event.

    - **nodes** – Where? Geographic nodes to target.

    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.

emodpy_measles.interventions.complex_import.**new_intervention_as_file**(*camp*, *timestep*,
                                                                        filename=None)

emodpy_measles.interventions.measles_vaccine module

emodpy_measles.interventions.measles_vaccine.**SimpleMCV1**(*camp*, *timestep*, *coverage=1.0*,
                                                            nodes=None)
    Create 'type 1' vaccine intervention campaign event at a specified time with a specified coverage in the specified
    nodes. Vaccine is distributed to infants 270 days after birth.

    Parameters

    - **timestep** – When? Timestep to distribute vaccine on.

- **coverage** – Who? The percentage of the population in the nodes who should receive the vaccine. Defaults to everyone if omitted.

- **nodes** – Where? The list of node ids where the vaccine should be distribution. Defaults to all if omitted.

Returns campaign event.

emodpy_measles.interventions.measles_vaccine.**SimpleMCV2**(*camp*, *timestep*, *coverage=1.0*, *nodes=None*)

Create 'type 2' vaccine intervention campaign event at a specified time with a specified coverage in the specified nodes. Vaccine is distributed to infants 475 days after birth.

Parameters

- **timestep** – When? Timestep to distribute vaccine on.

- **coverage** – Who? The percentage of the population in the nodes who should receive the vaccine. Defaults to everyone if omitted.

- **nodes** – Where? The list of node ids where the vaccine should be distribution. Defaults to all if omitted.

Returns campaign event.

emodpy_measles.interventions.measles_vaccine.**SingleDoseRI**(*camp*, *RI_MCV1*, *dose_dist=None*, *nodes=None*, *waning=1.0*, *start_day=0*, *take_by_age_ages=None*, *take_by_age_values=None*)

Single dose measles vaccine with a dosing delay distribution and take by age.

Parameters

- **camp** (`emod_api.campaign`) – The campaign instance

- **RI_MCV1** (`float`) – Demographic coverage of routine immunization in the range [0,1]

- **dose_dist** (`dict`, optional) – Dose delay distribution. Defaults to {'Delay_Period_Distribution': "GAUSSIAN_DISTRIBUTION", 'Delay_Period_Gaussian_Mean': 300.0, 'Delay_Period_Gaussian_Std_Dev': 90.0})

- **nodes** (`list`, `optional`) – Node ids where the vaccine should be distribution. Defaults to all.

- **waning** (`float`, `optional`) – Constant rate of dose efficacy (See WaningEffectConstant). Defaults to 1.0.

- **start_day** (`int`, `optional`) – Day to start the campaign. Defaults to 0.

- **take_by_age_ages** (`list`, `optional`) – List of ages in years for the vaccines take-by-age.

- **take_by_age_values** (`list`, `optional`) – List of take values for the vaccines take-by-age. Default is 0.85 < 9mo and 0.95 after.

Returns A `emod_api.schema_to_class.ReadOnlyDict` intervention

emodpy_measles.interventions.measles_vaccine.**new_intervention_as_file**(*camp*, *timestep*, *filename=None*)

Create a new 'campaign.json'-type file with a single campaign event containing a Measles Vaccine intervention. This is mostly a test or didactic capability.

Parameters

- **timestep** – When? Timestep to distribute vaccine on.

- **filename** – The name of the file to write to disk. Defaults to "measles_vaccine.json".

    Returns string with filename of newly created file.

emodpy_measles.interventions.simple_sia module

emodpy_measles.interventions.simple_sia.**SimpleSIA**(*camp*, *Event_Name='SIA'*, *timestep=1*,
                                          Target_Age_Min=0.75, *Target_Age_Max=5.0*,
                                          Coverage=1.0, *nodes=None*)

    Create and return a vaccine campaign event.

        Parameters

- **timestep** – When? Timestep at which vaccine event occurs. Defaults to 1.
- **Target_Age_Min** – Who (youngest)? Youngest age to distribute vaccine to (defaults to 9 months).
- **Target_Age_Max** – Who (oldest)? Oldest age to distribute vaccine to (defaults to 5yo).
- **Coverage** – Who (%age)? Percentage of population, in given age ranges, to get vaccine. Defaults to all.
- **nodes** – Where? Node ids where vaccine should be distributed. Defaults to everywhere.

        Returns New campaign event which can be added to campaign accumulator.

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

    Create a new 'campaign.json'-type file with a single campaign event containing a SimpleSIA style Measles vaccine intervention. This is mostly a test or didactic capability.

        Parameters

- **timestep** – When? Timestep to distribute vaccine on.
- **filename** – The name of the file to write to disk. Defaults to "sia.json".

        Returns string with filename of newly created file.

emodpy_measles.interventions.targeted_outbreak module

emodpy_measles.interventions.targeted_outbreak.**seed**(*camp*, *timestep*, *coverage=0.01*,
                                              node_ids=None)

    Seed an outbreak at a particular time, in selected nodes, based on coverage. This infects existing individuals.

        Parameters

- **camp** – emod_api.campaign 'module object'. Provides centralized source of schema.
- **timestep** – time(step) at which outbreak should be seeded.
- **coverage** – Percentage of individuals to be infected.
- **node_ids** – Nodes to target with outbreak. Leave empty to hit all nodes.

        Returns Configured campaign event ready to be added to campaign.

Submodules

emodpy_measles.campaign module

class emodpy_measles.campaign.**CampaignInfo**(*startday*, *agemin*, *agemax*, *region*, *coverage*, *label*)
  Bases: tuple

  property agemax
    Alias for field number 2

  property agemin
    Alias for field number 1

  property coverage
    Alias for field number 4

  property label
    Alias for field number 5

  property region
    Alias for field number 3

  property startday
    Alias for field number 0

emodpy_measles.campaign.**ConstructHistoricalCampaigns**()
  Building historical campaign calendar by hand from the WHO calendar, rather than trying to process that file, which has a free-form "comment field" with important info. Coverages from scenario files. Ignoring mopup and OBR, including the large OBR in NE in Jan 2017 Format (date, minimum age target, maximum age target, target region, target coverage, SIA ID)

DOCUMENTATION

Documentation for emodpy-measles should follow IDM's documentation toolchain.

PYTHON MODULE INDEX