
emodpy-malaria

Institute for Disease Modeling

Jan 28, 2022

CONTENTS

1	Installation prerequisites	3
1.1	emodpy-malaria installation	3
2	Frequently asked questions	5
2.1	What are some of the key differences for people used to using dtk-tools?	6
2.2	Do I need to install a whole bunch of Python modules? Where do I get that list?	6
2.3	Is there an easier way than typing <code>--index-url</code> with that long URL every time I use pip?	7
2.4	How do I find the path or file to my <code>pip.ini</code> ?	7
2.5	What if the <code>pip.ini</code> file (or <code>pip</code> folder) doesn't exist?	7
2.6	What's the text I need to enter into my <code>pip</code> config file?	7
2.7	I was installing the Python modules and it told me I needed a C++ compiler?	7
2.8	I was installing and got prompted for JFrog credentials.	7
2.9	What if I really actually do want to install something from staging?	8
2.10	What version of Python should I be using?	8
2.11	What if I want a particular version of <code>emodpy-malaria</code> ?	8
2.12	What if I want to make changes to <code>emodpy-malaria</code> and run with those, instead of a released version?	8
2.13	I pip installed <code>emodpy-malaria</code> , but I want to make changes. How should I do that?	8
2.14	How do I set configuration parameters?	9
2.15	How do I specify the log level for EMOD? I get a schema error when I try to set it now.	9
2.16	How do I specify the vector species for my scenario?	10
2.17	Is there an example of creating a demographics file from scratch with the API?	10
2.18	I see a lot of <code>MALARIA_SIM</code> examples. Are there any <code>VECTOR_SIM</code> examples?	11
2.19	Is there a multi-node or spatial example?	11
2.20	Are there simple campaign/intervention examples?	11
2.21	Is there a drug campaign example?	11
2.22	Is there a campaign sweep example?	12
2.23	Is there a demographics sweep example?	12
2.24	Is there a serialization/burn-in example?	12
2.25	Is there a reporter configuration example?	12
3	emodpy_malaria	13
3.1	emodpy_malaria package	13
3.1.1	Subpackages	13
3.1.2	Submodules	79
	Python Module Index	89
	Index	91

emodpy-malaria is a collection of Python scripts and utilities created to streamline user interactions with EMOD and idmtools for modeling malaria. Much of the functionality is inherited from the [emod_api](#) and [emodpy](#) packages.

Additional information about how to use idmtools can be found in [Welcome to idmtools](#). Additional information about EMOD malaria parameters 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.

INSTALLATION PREREQUISITES

First, ensure the following prerequisites are met before following the install instructions in *emodpy-malaria installation*.

- Windows 10 Pro or Enterprise, or Linux
- Python 3.6.2 or 3.7 64-bit (<https://www.python.org/downloads/release>)
- Git client, such as Git Bash or the Git GUI
- pip.ini (Windows) or pip.conf (Linux), containing the following:

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

1.1 emodpy-malaria installation

Follow the steps below to install emodpy-malaria.

Note: Currently, VPN connection is required to run the example.

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

```
python -m venv v-emodpy-malaria
```

2. Activate the virtual environment:

- Windows
- Linux

Enter the following:

```
v-emodpy-malaria\Scripts\activate
```

Enter the following:

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

3. Install emodpy-malaria packages:

```
pip install emodpy_malaria
```

If you are on Python 3.6, also run:

```
pip install dataclasses
```

If you are on Linux, also run:

```
pip install keyrings.alt
```

4. Open a command prompt and clone the emodpy-malaria GitHub repository to a local directory using the following command:

```
git clone https://github.com/InstituteForDiseaseModeling/emodpy-malaria.git
```

5. Verify installation by running the included Python example, `example.py`, located in `/examples/start_here`:

```
python example.py
```

Upon completion you can view the results in COMPS.

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

```
deactivate
```


FREQUENTLY ASKED QUESTIONS

As you get started with emodpy-malaria, you may have questions. The most common questions are answered below. The most common questions are answered below. For questions related to functionality in related packages, see [Frequently asked questions](#) for idmtools, [Frequently asked questions](#) for emod-api, and [Frequently asked questions](#) for emodpy.

Contents

- *What are some of the key differences for people used to using dtk-tools?*
- *Do I need to install a whole bunch of Python modules? Where do I get that list?*
- *Is there an easier way than typing `--index-url` with that long URL every time I use pip?*
- *How do I find the path or file to my pip.ini?*
- *What if the pip.ini file (or pip folder) doesn't exist?*
- *What's the text I need to enter into my pip config file?*
- *I was installing the Python modules and it told me I needed a C++ compiler?*
- *I was installing and got prompted for JFrog credentials.*
- *What if I really actually do want to install something from staging?*
- *What version of Python should I be using?*
- *What if I want a particular version of emodpy-malaria?*
- *What if I want to make changes to emodpy-malaria and run with those, instead of a released version?*
- *I pip installed emodpy-malaria, but I want to make changes. How should I do that?*
- *How do I set configuration parameters?*
- *How do I specify the log level for EMOD? I get a schema error when I try to set it now.*
- *How do I specify the vector species for my scenario?*
- *Is there an example of creating a demographics file from scratch with the API?*
- *I see a lot of MALARIA_SIM examples. Are there any VECTOR_SIM examples?*
- *Is there a multi-node or spatial example?*
- *Are there simple campaign/intervention examples?*
- *Is there a drug campaign example?*
- *Is there a campaign sweep example?*

- *Is there a demographics sweep example?*
- *Is there a serialization/burn-in example?*
- *Is there a reporter configuration example?*

2.1 What are some of the key differences for people used to using dtk-tools?

1. Schema-Based. The creation of config and campaign files is entirely schema-based now. This means that you can only set parameters that the binary you are using recognizes. And parameter types and ranges are enforced at runtime.
2. Inter-File Dependencies Now Automatic. Before there were lots of parameters in the config that you had to set to correspond to settings in campaign or demographics files. That is no longer the case. We call these ‘implicit’. For example, if you add a BirthRate to the demographics, the corresponding parameters in the config.json (Enable_Births) will get set automatically for you. As another example, when you create a campaign and specify various ‘events’ to be broadcast/published and/or listened/subscribed to, you no longer have to figure out which ones are built-in and which are ad-hoc. It does that for you and populates the Custom_Events param on your behalf.
3. Hierarchical Dependencies Now Automatic. If a parameter depends on another parameter, previously you had to set all the Enables in the dependency tree. Now they get set automatically for you. For example, if Enable_Birth is set (see above), Enable_Vital_Dynamics will be set for you.
4. No JSON manipulation. dtk-tools worked primarily through manipulation of JSON that made up the configuration files. You no longer need to have any knowledge of the internal representation of data in the DTK input files. All configuration should be done via Python functions.
5. Released and Installed Modules. We want users mostly using versioned released modules that have been pip installed, not git cloned, dev-installed code, except during development. The process of getting new code reviewed, tested, and getting the module versioned and released is intended to be smooth and efficient when everyone does their defined role. “Trust The Process and Do Your Job”, as someone once said.
6. Blessed Binaries. In dtk-tools you would often BYOB – Bring Your Own Binary – but in emodpy, the idea is that the system pulls down the latest CI (Continuous Integration) build for your disease that passed all the tests. We very much want to normalize the idea of doing research with versioned software that has come through our professional BVT processes.

2.2 Do I need to install a whole bunch of Python modules? Where do I get that list?

No. You should only have to install emodpy-malaria, and everything else should come as a dependency automatically. If you are starting from a workflow repo, there should be a requirements.txt file that lists a particular version of emodpy-malaria. Then ‘pip install -r requirements.txt’ is what you should expect to do.

2.3 Is there an easier way than typing `--index-url` with that long URL every time I use pip?

Yes. You should not be typing that every time. You should update the `pip.ini` (Windows) or `pip.conf` globally on your computer with that URL and never have to type it again.

2.4 How do I find the path or file to my `pip.ini`?

```
pip config -v list.
```

Choose the first or second one.

2.5 What if the `pip.ini` file (or `pip` folder) doesn't exist?

Go ahead and create it (at one of the locations specified by `'pip config -v list'`). It's your computer.

2.6 What's the text I need to enter into my `pip config` file?

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

2.7 I was installing the Python modules and it told me I needed a C++ compiler?

This is because of a dependency in some versions of `emod-api` on `lz4`. `lz4` usually comes as what's called a source package and it has to be compiled as part of the install. This requires a compiler, which doesn't come by default on Windows computers. At this point, if your `pip.ini` is set up properly, you should be able to get a pre-built `lz4` package from our local Artifactory PIP server.

2.8 I was installing and got prompted for JFrog credentials.

You should not have to give credentials to use our JFrog/Artifactory/pip server. Credentials are required to get packages from the staging server. We don't expect end users to be accessing packages from staging, just prod, which is auth-free.

2.9 What if I really actually do want to install something from staging?

You need to specify ‘`--index-url = https://<username>@idmod.org:<ssh...password>@packages.idmod.org/api/pypi/pypi-staging/simple`’ and also provide your creds when prompted.

2.10 What version of Python should I be using?

At least Python 3.7.7. If you are installing a new version of Python, feel free to go all the way forward to a Python 3.9.x. 3.8 is probably the sweet spot of “known to work and still not considered old”.

2.11 What if I want a particular version of emodpy-malaria?

```
pip install emodpy-malaria==1.2.3
```

Should get you what you need.

2.12 What if I want to make changes to emodpy-malaria and run with those, instead of a released version?

(This is a duplicate.)

There are a couple of ways of doing that. Option 1: Do a Dev Install:

```
pip install -e .
```

This will make your site packages map to your local code until you do a new pip install of the package.

Option 2: Creating a wheel from your local code and pip install it (each time you make a change).:

```
python setup.py bdist_wheel
pip3 install dist/<newly_create_file.whl>
```

Some people prefer option 1 because it’s “one and done”. Some people prefer option 2 because it keeps you thinking in terms of packaging, versioning, and installing even while you’re developing.

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

Install at a command prompt using the following:

```
python package_setup.py develop
```

This method is the most popular and proven, though there are some other options. Installing this way means that the emodpy-malaria module in site-packages actually points to the same code as you have checked out in git. For more detail, see this [Stack Overflow post](#).

However, we aim to get the desired changes quickly tested and included in the versioned module we release via pip install.

2.14 How do I set configuration parameters?

Define your own parameter-setting function such as `set_param_fn` and pass that function to the emodpy task creator as the `param_custom_cb` parameter. In that function, you can set the parameters directly. For example:

```
print("nSims: ", params.nSims)

def set_param_fn(config):
    """
    This function is a callback that is passed to emod-api.config to set parameters.
    ↪The Right Way.
    """
    import emodpy_malaria.malaria_config as conf
    config = conf.set_team_defaults(config, manifest)
    conf.add_species(config, manifest, ["gambiae"])

    return config

def build_campaign():
    """
    Build a campaign input file for the DTK using emod_api.
    Right now this function creates the file and returns the filename. If calling
    ↪code just needs an asset that's fine.
    """
    import emod_api.campaign as camp
    import emodpy_malaria.interventions.bednet as bednet
    import emodpy_malaria.interventions.treatment_seeking as ts

    # This isn't desirable. Need to think about right way to provide schema (once)
    camp.schema_path = manifest.schema_file

    # print( f"Telling emod-api to use {manifest.schema_file} as schema." )
    # camp.add( bednet.Bednet( camp, start_day=100, coverage=1.0, killing_eff=1.0,
```

See examples/start_here/example.py. for additional information.

If you prefer something more modular, you can call a function in a standalone script/file that sets the configuration parameters.

Are there defaults? Great question. If you don't set any configuration parameters, they will have defaults based on the schema. The malaria team has set team defaults in `emodpy_malaria.config.set_team_defaults()`. These defaults can be seen in `config.py`.

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

TBD

2.16 How do I specify the vector species for my scenario?

See the excerpt below or the complete example of setting the vector species and parameter values associated with each species in `examples/start_here/example.py`.

```
print("nSims: ", params.nSims)

def set_param_fn(config):
    """
    This function is a callback that is passed to emod-api.config to set parameters.
    ↪The Right Way.
    """
    import emodpy_malaria.malaria_config as conf
    config = conf.set_team_defaults(config, manifest)
    conf.add_species(config, manifest, ["gambiae"])

    return config

def build_campaign():
    """
    Build a campaign input file for the DTK using emod_api.
    Right now this function creates the file and returns the filename. If calling
    ↪code just needs an asset that's fine.
    """
    import emod_api.campaign as camp
    import emodpy_malaria.interventions.bednet as bednet
    import emodpy_malaria.interventions.treatment_seeking as ts

    # This isn't desirable. Need to think about right way to provide schema (once)
    camp.schema_path = manifest.schema_file

    # print( f"Telling emod-api to use {manifest.schema_file} as schema." )
    # camp.add( bednet.Bednet( camp, start_day=100, coverage=1.0, killing_eff=1.0,
```

A helper function to make this task even easier may be coming shortly.

2.17 Is there an example of creating a demographics file from scratch with the API?

The best examples are currently in `emodpy-measles` and `emodpy-hiv`. We are working to add some to `emod_api.demographics`. The basic idea is you use one of 3 node creators, and then use the Setter API to set up the node defaults for fertility, mortality, age structure, initial immunity, individual 'risk', and initial prevalence. The first node creator, `from_template_node`, is very basic and usually for quickstarts or toy models. It lets you create a single node demographics file with a given population. The second creator, `from_csv`, lets you create a multinode demographics using a csv file with population data as an input. The third creator, `from_params`, lets you create a multinode demographics without specific node data but instead with a few parameters that represent the overall population and the population heterogeneity.

This is what it could look like to use option 2:

```
from emodpy_malaria.demographics import MalariaDemographics as Demographics
demog = Demographics.from_csv( input_csv_file )
```

(continues on next page)

(continued from previous page)

```
demog.SetConstantRisk()  
demog.SetInitialAgeLikeSubSaharanAfrica()  
demog.generate_file(out_filename)  
return demog
```

2.18 I see a lot of MALARIA_SIM examples. Are there any VECTOR_SIM examples?

Yes! The following examples use VECTOR_SIM:

- examples/migration_spatial_vector_sim
- examples/vector_basic
- examples/vector_genetics_insecticide_resistance
- examples/vector_genetics_vector_sim

2.19 Is there a multi-node or spatial example?

Yes. See:

- examples/migration_spatial_vector_sim
- examples/migration_spatial_malaria_sim
- examples/jonr_1

2.20 Are there simple campaign/intervention examples?

Yes. See:

- examples/outdoor_rest_kill_male_mosquitoes
- examples/inputEIR
- examples/ivermectin

2.21 Is there a drug campaign example?

Yes. See:

- examples/drug_campaign
- examples/diagnostic_survey

2.22 Is there a campaign sweep example?

Yes. See:

- [examples/campaign_sweep](#)

2.23 Is there a demographics sweep example?

Yes. See:

- [examples/demographics_sweep](#)

2.24 Is there a serialization/burn-in example?

Yes. See:

- [examples/burnin_create](#)
- [examples/burnin_use](#)

2.25 Is there a reporter configuration example?

Yes. See:

- [examples/add_reports](#)
- [examples/filtered_report](#)

EMODPY_MALARIA

3.1 emodpy_malaria package

3.1.1 Subpackages

`emodpy_malaria.demographics` package

Submodules

`emodpy_malaria.demographics.MalariaDemographics` module

This module contains the classes and functions for creating demographics files for malaria simulations. For more information on EMOD demographics files, see [Demographics file](#).

```
class emodpy_malaria.demographics.MalariaDemographics(nodes,  
                                                    idref='Gridded  
                                                    world  
                                                    grump2.5arcmin',  
                                                    base_file=None,  
                                                    init_prev=0.0,  
                                                    in-  
                                                    clude_biting_heterogene
```

Bases: `emod_api.demographics.Demographics.Demographics`

This class is derived from `emod_api.demographics.Demographics.Demographics` and sets certain defaults for malaria in construction.

Parameters

- **nodes** – The number of nodes to create.
- **idref** – 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](#).
- **base_file** – A basic demographics file used as a starting point for creating more complicated demographics files. For example, using a single node file to create a multi-node file for spatial simulations.
- **init_prev** – The initial malaria prevalence of the population. Defaults to 0%.
- **include_biting_heterogeneity** – variable biting rates. Defaults to on.

Returns None

set_risk_lowmedium()

Set initial risk for low-medium transmission settings per: <https://wiki.idmod.org/display/MAL/Heterogeneous+biting+risk+in+simulations+vs+data>.

set_risk_high()

Set initial risk for high transmission settings per: <https://wiki.idmod.org/display/MAL/Heterogeneous+biting+risk+in+simulations+vs+data>.

add_larval_habitat_multiplier(*schema, hab_type, multiplier, species='ALL_SPECIES', node_id=0*)

Add LarvalHabitatMultiplier to node(s).

Parameters

- **schema** – Path to schema.json.
- **hab_type** – Habitat type.
- **multiplier** – Multiplier or Factor.
- **specices** – Specific species (defaults to ALL).
- **node_id** – Nodes for this LHM. Defaults to all.

Returns Nothing.

```
emodpy_malaria.demographics.MalariaDemographics.from_template_node(lat=0,
                                                                    lon=0,
                                                                    pop=1000000.0,
                                                                    name=1,
                                                                    forced_id=1,
                                                                    init_prev=0.2,
                                                                    in-
                                                                    clude_biting_heterogeneity=True)
```

Create a single-node *MalariaDemographics* instance from the parameters you supply.

Parameters

- **lat** – Latitude of the centroid of the node to create.
- **lon** – Longitude of the centroid of the node to create.
- **pop** – Human population of the node.
- **name** – The name of the node. This may be a characteristic of the node, such as “rural” or “urban”, or an identifying integer.
- **forced_id** – The node ID for the single node.
- **init_prev** – The initial malaria prevalence of the node.

Returns A *MalariaDemographics* instance.

```
emodpy_malaria.demographics.MalariaDemographics.from_pop_csv(pop_filename_in,
                                                                pop_filename_out='spatial_gridded_pop_di',
                                                                site='No_Site')
```

Create a multi-node *MalariaDemographics* 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_malaria.demographics.MalariaDemographics.from_csv(input_file,
                                                           res=0.008333333333333333,
                                                           id_ref='from_csv',
                                                           init_prev=0.0,          in-
                                                           clude_biting_heterogeneity=True)
```

Create a multi-node *MalariaDemographics* instance from a CSV file describing a population.

Parameters

- **input_file** – The path to the csv file to ingest.
- **res** – Resolution.
- **id_ref** – A string to identify the file, needs to match other input files.
- **init_prev** – The initial malaria prevalence of the population. Defaults to 0%.
- **include_biting_heterogeneity** – variable biting rates. Defaults to on.

Returns A *MalariaDemographics* instance

```
emodpy_malaria.demographics.MalariaDemographics.from_params(tot_pop=1000000.0,
                                                             num_nodes=100,
                                                             frac_rural=0.3,
                                                             id_ref='from_params')
```

Create a multi-node *MalariaDemographics* 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 *MalariaDemographics* instance.

emodpy_malaria.interventions package

Submodules

emodpy_malaria.interventions.bednet module

This module contains functionality for bednet distribution.

```
emodpy_malaria.interventions.bednet.BednetIntervention(schema_path, blocking_eff:
    float = 0.9, killing_eff: float
    = 0.6, repelling_eff: float
    = 0, usage_eff: float =
    1, blocking_decay_rate:
    float = 0, blocking_predecay_duration: int
    = 730, killing_decay_rate:
    float = 0, killing_predecay_duration:
    int = 1460, repelling_decay_rate:
    float = 0, repelling_predecay_duration:
    int = 365, usage_decay_rate: float = 0,
    usage_predecay_duration:
    int = 365, insecticide: str =
    None, cost: float = 0)
```

To define the durability of the bednets. :param *schema_path*: Path to the schema :param *blocking_eff*: Blocking efficacy :param *killing_eff*: Killing efficacy :param *repelling_eff*: Repelling efficacy :param *usage_eff*: Usage efficacy :param *blocking_decay_rate*: Blocking decay rate :param *blocking_predecay_duration*: Blocking predecay duration :param *killing_decay_rate*: Killing decay rate :param *killing_predecay_duration*: Killing Predecay :param *repelling_decay_rate*: Repelling decay rate :param *repelling_predecay_duration*: Repelling predecay duration :param *usage_decay_rate*: Usage decay rate :param *usage_predecay_duration*: Usage predecay duration :param *insecticide*: The name of the insecticide used in the bednet. :param *cost*: Cost for the intervention.

Returns:

```
emodpy_malaria.interventions.bednet.Bednet(schema_path, start_day: int = 0, coverage:
    float = 1.0, blocking_eff: float = 1,
    killing_eff: float = 1, repelling_eff: float =
    1, usage_eff: float = 1, blocking_decay_rate:
    float = 0, blocking_predecay_duration:
    int = 365, killing_decay_rate: float
    = 0, killing_predecay_duration: int
    = 365, repelling_decay_rate: float =
    0, repelling_predecay_duration: int =
    365, usage_decay_rate: float = 0, usage_predecay_duration: int = 365, node_ids:
    list = None, insecticide: str = None)
```

Add a simple insecticide-treated bednet intervention with configurable efficacy and usage that can decay over time to your campaign. This is equivalent to [UsageDependentBednet](#) with exponential waning.

Parameters

- ***schema_path*** – path to the schema
- ***start_day*** – The day of the simulation on which the bednets are distributed. We recommend aligning this with the start of the simulation.
- ***coverage*** – The proportion of the population that will receive a bednet.
- ***blocking_eff*** – The efficacy of the bednet at blocking mosquitoes from feeding.
- ***killing_eff*** – The efficacy of the bednet at killing mosquitoes that land on it.
- ***repelling_eff*** – The efficacy of the bednet at repelling mosquitoes.

- **usage_eff** – The proportion of time that individuals with a bednet use it. This effectively reduces the other efficacy measures by the amount specified; a value of 0.5 will reduce blocking, killing, and repelling efficacies by half.
- **blocking_decay_rate** – The exponential decay length, in days, where the current effect is equal initial efficacy - dt/decay rate.
- **blocking_predecay_duration** – The time, in days, before bednet efficacy begins to decay.
- **killing_decay_rate** – The exponential decay length, in days, where the current effect is equal initial efficacy - dt/decay rate.
- **killing_predecay_duration** – The time, in days, before bednet efficacy begins to decay.
- **repelling_decay_rate** – The exponential decay length, in days, where the current effect is equal initial efficacy - dt/decay rate.
- **repelling_predecay_duration** – The time, in days, before bednet efficacy begins to decay.
- **usage_decay_rate** – The exponential decay length, in days, where the current usage is equal initial efficacy - dt/decay rate.
- **usage_predecay_duration** – The time, in days, before bednet usage begins to decay.
- **node_ids** – The IDs of the nodes in which to distribute the bednets.
- **insecticide** – The name of the insecticide used in the bednet.

Returns The bednet intervention event.

```
emodpy_malaria.interventions.bednet.BasicBednet(camp, start_day, coverage=1.0,
                                                  blocking_eff=1, killing_eff=1, re-
                                                  pelling_eff=1, usage_eff=1, insecti-
                                                  cide=None)
```

Add a simpler insecticide-treated bednet intervention with constant efficacy and usage to your campaign. This is equivalent to [SimpleBednet](#).

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **start_day** – The day of the simulation on which the bednets are distributed. We recommend aligning this with the start of the simulation.
- **coverage** – The proportion of the population that will receive a bednet.
- **blocking_eff** – The efficacy of the bednet at blocking mosquitoes from feeding.
- **killing_eff** – The efficacy of the bednet at killing mosquitoes that land on it.
- **repelling_eff** – The efficacy of the bednet at repelling mosquitoes.
- **usage_eff** – The proportion of individuals with a bednet who use it.
- **insecticide** – The insecticide used in the bednet.

Returns The bednet intervention event.

```
emodpy_malaria.interventions.bednet.new_intervention_as_file(camp, start_day,
                                                             filename=None)
```

Write a campaign file to disk with a single bednet event, using defaults. Useful for testing and learning.

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **start_day** – The day of the simulation on which the bednets are distributed. We recommend aligning this with the start of the simulation.
- **filename** – The campaign filename; can be omitted and default will be used and returned to user.

Returns The campaign filename written to disk.

```
emodpy_malaria.interventions.bednet.add_ITN_scheduled(camp, start_day: int = 0, coverage_by_ages: list = None, node_ids: list = None, itn_bednet: emodpy_malaria.interventions.bednet.BednetIntervention = None, blocking_eff: float = 0.9, blocking_predecay_duration: int = 0, blocking_decay_rate: float = 0.000136986301369863, killing_eff: float = 0.6, killing_predecay_duration: int = 0, killing_decay_rate: float = 0.000136986301369863, ind_property_restrictions: list = None, repetitions: int = 1, tsteps_btwn_repetitions: int = 1095, receiving_itn_event_name: str = "")
```

Add a scheduled bednet intervention.

Parameters

- **camp** – object for building, modifying, and writing campaign configuration files.
- **start_day** – Start day of intervention.
- **coverage_by_ages** – A list of dictionaries defining the coverage per age group. For example, [{"coverage":1,"min": 1, "max": 10}, {"coverage":1, "min": 11, "max": 50}, {"coverage":0.5, "birth":"birth", "duration":34}].
- **node_ids** – The list of nodes to apply this intervention to. If not provided, set value of `NodeSetAll`.
- **itn_bednet** – An object of class `Bednet` to define the durability of the nets. If not provided, the default decay profile of a `Bednet` intervention is used.
- **blocking_eff** – Initial blocking efficacy.
- **blocking_predecay_duration** – How long blocking efficacy holds prior to decay.
- **blocking_decay_rate** – Blocking decay rate after constant period.
- **killing_eff** – Initial killing efficacy.
- **killing_predecay_duration** – How long killing efficacy holds prior to decay.
- **killing_decay_rate** – Killing decay rate after constant period.

- **ind_property_restrictions** – The IndividualProperty key:value pairs that individuals must have to receive the intervention (**Property_Restrictions_Within_Node** parameter). In the format [{ "BitingRisk": "High"}, {"IsCool": "Yes"}].
- **repetitions** – Repeat this intervention **repetitions** times.
- **tsteps_btwn_repetitions** – Number of timesteps between repetitions.
- **receiving_itn_event_name** – Name of event/trigger/signal to publish when bednet is distributed.

Returns None

```
emodpy_malaria.interventions.bednet.add_ITN_triggered(camp, start_day: int = 0, event_name: str = "", coverage_by_ages: list = None, itn_bednet: emodpy_malaria.interventions.bednet.Bednet = None, node_ids: list = None, insecticide: str = None, node_property_restrictions: list = None, triggered_campaign_delay: int = None, trigger_condition_list: list = None, listening_duration: int = -1, check_eligibility_at_trigger: bool = False, repetitions: int = 1, tsteps_btwn_repetitions: int = 1095)

emodpy_malaria.interventions.bednet.add_ITN(camp, start: int = 0, coverage_by_ages: list = None, itn_bednet: emod_api.schema_to_class.ReadOnlyDict = None, nodeIDs: list = None, insecticide: str = None, node_property_restrictions: list = None, ind_property_restrictions: list = None, triggered_campaign_delay: int = None, trigger_condition_list: list = None, listening_duration: int = -1, check_eligibility_at_trigger: bool = False, repetitions: int = 1, tsteps_btwn_repetitions: int = 1095)
```

Add an insecticide-treated net (ITN) intervention to the campaign using the **SimpleBednet** class.

Parameters

- **camp** – The DTKConfigBuilder containing the campaign configuration.
- **start** – The day on which to start distributing the bednets (**Start_Day** parameter).
- **coverage_by_ages** – A list of dictionaries defining the coverage per age group. For example, [{"coverage":1, "min": 1, "max": 10}, {"coverage":1, "min": 11, "max": 50}, {"coverage":0.5, "birth":"birth", "duration":34}].
- **itn_bednet** – A dictionary defining the durability of the nets. If not provided, the default decay profile for **Killing_Config**, **Blocking_Config**, and **Usage_Config** are used. For example, to update usage duration, provide {"Usage_Config" : {"Expected_Discard_Time": 180}}.

- **cost** – The per-unit cost (**Cost_To_Consumer** parameter).
- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of `NodeSetAll`.
- **insecticide** – name of insecticide to use, must be in `config.json` Default `None`.
- **ind_property_restrictions** – The `IndividualProperty` key:value pairs that individuals must have to receive the intervention (**Property_Restrictions_Within_Node** parameter). In the format `[{ "BitingRisk": "High"}, { "IsCool": "Yes"}]`.
- **node_property_restrictions** – The `NodeProperty` key:value pairs that nodes must have to receive the intervention (**Node_Property_Restrictions** parameter). In the format `[{ "Place": "RURAL"}, { "ByALake": "Yes"}]`.
- **triggered_campaign_delay** – After the trigger is received, the number of time steps until the campaign starts. Eligibility of people or nodes for the campaign is evaluated on the start day, not the triggered day.
- **trigger_condition_list** – (Optional) A list of the events that will trigger the ITN intervention. If included, **start** is the day when monitoring for triggers begins. This argument cannot configure birth-triggered ITN (use **coverage_by_ages** instead).
- **listening_duration** – The number of time steps that the distributed event will monitor for triggers. Default is -1, which is indefinitely.
- **check_eligibility_at_trigger** – if triggered event is delayed, you have an option to check individual/node's eligibility at the initial trigger or when the event is actually distributed after delay.

Returns None

Example

```
camp = DTKConfigBuilder.from_defaults(sim_example) coverage_by_ages = [{"coverage": 1,
"min": 1, "max": 10}, {"coverage": 0.75, "min": 11, "max": 60}] add_ITN(camp,
start=1, coverage_by_ages, nodeIDs=[2, 5, 7, 21], node_property_restrictions=[{"Place": "Urban"}],
ind_property_restrictions=[{"Biting_Risk": "High"}], triggered_campaign_delay=14, trigger_condition_list=["NewClinicalCase", "NewSevereCase"], listening_duration=-1)
```

emodpy_malaria.interventions.common module

This module contains functionality common to many interventions.

```
emodpy_malaria.interventions.common.MalariaDiagnostic(camp, Diagnostic_Type='BLOOD_SMEAR_PARASITES',
Measurement_Sensitivity=0, Detection_Threshold=0)
```

Add a malaria diagnostic intervention to your campaign. This is equivalent to [MalariaDiagnostic](#).

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **Measurement_Sensitivity** – The setting for **Measurement_Sensitivity** in [MalariaDiagnostic](#).
- **Detection_Threshold** – The setting for **Detection_Threshold** in [MalariaDiagnostic](#).

- **Diagnostic_Type** – The setting for **Diagnostic_Type** in `MalariaDiagnostic`. In addition to the accepted values listed there, you may also set `TRUE_INFECTION_STATUS`, which calls `StandardDiagnostic` instead.

Returns The diagnostic intervention event.

`emodpy_malaria.interventions.common.AntimalarialDrug(camp, Drug_Type, ctc=1.0)`

Add an antimalarial drug intervention to your campaign. This is equivalent to `AntimalarialDrug`.

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **Drug_Type** – The name of the drug to distribute in a drugs intervention. Possible values are contained in `Malaria_Drug_Params`.
- **ctc** – The cost to consumer.

Returns The antimalarial drug intervention event.

`emodpy_malaria.interventions.common.triggered_campaign_delay_event(camp, start_day, trigger, delay, intervention, ip_targeting=None, coverage=1.0, target_age_min=0, target_age_max=125.0)`

Create and return a campaign event that responds to a trigger after a delay with an intervention.

Parameters

- **camp** – `emod_api.campaign` object with `schema_path` set.
- **start_day** – When to start.
- **trigger** – E.g., “NewInfection”.
- **delay** – Dictionary of 1 or 2 params that are the literal `Delay_Distribution` parameters, but without the distribution, which is inferred. E.g., { “Delay_Period_Exponential”: 5 }.
- **intervention** – List of 1 or more valid intervention dictionaries to be distributed together.
- **ip_targeting** – Optional Individual Properties required for someone to receive the intervention(s).
- **coverage** – Optional percentage coverage of target population. Defaults to all (100%).
- **target_age_min** – Optional minimum age (in years). Defaults to 0.
- **target_age_max** – Optional maximum age (in years). Defaults to 125.

Returns Campaign event.

emodpy_malaria.interventions.community_health_worker module

```
emodpy_malaria.interventions.community_health_worker.add_community_health_worker (campaign,
start_day:
    int
    =
    1,
    trig-
    ger_condition:
    list
    =
    None,
    de-
    mo-
    graphic_coverage:
    float
    =
    1.0,
    node_ids:
    list
    =
    None,
    ind_property:
    list
    =
    None,
    node_property:
    list
    =
    None,
    tar-
    get_age_min:
    int
    =
    0,
    tar-
    get_age_max:
    int
    =
    125,
    tar-
    get_gender:
    str
    =
    'All',
    ini-
    tial_amount:
    int
    =
    6,
    amount_in_sh:
    int
    =
    2147480000,
    days_between:
    float
    =
    7,
    du-
    ra-
```

Sets up a `CommunityHealthWorkerEventCoordinator` with the passed in intervention

Parameters

- **campaign** – campaign object to which the intervention will be added, and `schema_path` container
- **start_day** – The day the intervention is given out.
- **trigger_condition_list** – The list of individual events that are of interest to the community health worker (CHW). If one of these events occurs, the individual or node is put into a queue to receive the CHW's intervention. The CHW processes the queue when the event coordinator is updated.
- **demographic_coverage** – This value is the probability that each individual in the target population will receive the intervention. It does not guarantee that the exact fraction of the target population set by `Demographic_Coverage` receives the intervention.
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **ind_property_restrictions** – A list of dictionaries of `IndividualProperties`, which are needed for the individual to receive the intervention. Sets the **Property_Restrictions_Within_Node**
- **node_property_restrictions** – A list of the `NodeProperty` key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**
- **target_age_min** – The lower end of ages targeted for an intervention, in years. Sets **Target_Age_Min**
- **target_age_max** – The upper end of ages targeted for an intervention, in years. Sets **Target_Age_Max**
- **target_gender** – The gender targeted for an intervention: All, Male, or Female.
- **initial_amount** – Each instance will receive this constant/fixed value. Uses **Initial_Amount_Constant**
- **amount_in_shipment** – The number of interventions (such as vaccine doses) that a health worker or clinic receives in a shipment.
- **days_between_shipments** – The number of days to wait before a clinic or health worker receives a new shipment of interventions (such as vaccine doses)
- **duration** – The number of days for an event coordinator to be active before it expires. -1 means it never expires.
- **intervention_config** – A configured intervention to be distributed by the coordinator
- **max_distributed_per_day** – The maximum number of interventions (such as vaccine doses) that can be distributed by health workers or clinics in a given day
- **max_stock** – The maximum number of interventions (such as vaccine doses) that can be stored by a health worker or clinic
- **waiting_period** – The number of days a person or node can be in the queue waiting to get the intervention from the community health worker (CHW)

Returns Nothing

emodpy_malaria.interventions.diag_survey module

This module contains functionality for diagnostic survey interventions.

```
emodpy_malaria.interventions.diag_survey.add_diagnostic_survey(camp, coverage:
    float = 1, repetitions: int = 1,
    tsteps_btwn_repetitions: int = 365, target: object
    = 'Everyone', start_day: int
    = 1, diagnostic_type: str =
    'BLOOD_SMEAR_PARASITES', diagnos-
    tic_threshold:
    float = 40, measure-
    ment_sensitivity:
    float = 0.1, event_name:
    str = 'Diagnostic Survey',
    node_ids: list
    = None, positive_diagnosis_configs:
    list =
    None, negative_diagnosis_configs:
    list = None, received_test_event:
    str = 'Received_Test',
    IP_restrictions:
    list = None,
    NP_restrictions:
    list = None,
    disqualify-
    ing_properties:
    list = None, trigger_condition_list:
    list = None,
    listening_duration:
    int = -1, triggered_campaign_delay:
    int = 0,
    check_eligibility_at_trigger:
    bool = False, expire_recent_drugs:
    any = None)
```

Add an intervention to create either a scheduled or a triggered event to the campaign using the `MalariaDiagnostic` class, an individual-level class, to test individuals. Upon positive or negative diag-

nosis, the list of events to occur (as defined in **positive_diagnosis_configs** or **negative_diagnosis_configs**) is then executed. These events can trigger other listening interventions.

Parameters

- **camp** – The `emod_api.campaign` object for building, modifying, and writing campaign configuration files.
- **coverage** – The probability an individual receives the diagnostic.
- **repetitions** – Number of repetitions of the survey intervention.
- **tsteps_btwn_repetitions** – Timesteps between repetitions.
- **target** – A dictionary targeting an age range and gender of individuals for treatment. In the format {"agemin": x, "agemax": y, "gender": z}. Default is Everyone.
- **start_day** – The day of the simulation on which the intervention is created. If triggered, runs on trigger, not on **start_day**.
- **diagnostic_type** – Type of malaria diagnostic used. Default is **BLOOD_SMEAR_PARASITES**. Available options are:
 - **BLOOD_SMEAR_PARASITES**
 - **BLOOD_SMEAR_GAMETOCYTES**
 - **PCR_PARASITES**
 - **PCR_GAMETOCYTES**
 - **PF_HRP2**
 - **TRUE_INFECTION_STATUS**
 - **TRUE_PARASITE_DENSITY**
 - **FEVER**
- **diagnostic_threshold** – The diagnostic detection threshold based on **diagnostic_type**:
 - TRUE_INFECTION_STATUS**
 - BLOOD_SMEAR_PARASITES** In parasites/microliter, use measurement = float(1.0 / measurementSensitivity * Poisson(measurementSensitivity * true_parasite_density)).
 - BLOOD_SMEAR_GAMETOCYTES** In gametocytes/microliter, use measurement = float(1.0 / measurementSensitivity * Poisson(measurementSensitivity * true_gametocyte_density)).
 - PCR_PARASITES and PCR_GAMETOCYTES** Use the true values and an algorithm based on the paper [Improving statistical inference on pathogen densities estimated by quantitative molecular methods : malaria gametocytaemia as a case study](#).
 - PF_HRP2** Add a new method to get the PfHRP2 value and check against the threshold.
 - TRUE_PARASITE_DENSITY** Check the true parasite density against the threshold.
 - FEVER** Check the person's fever against the threshold.
- **measurement_sensitivity** – Setting for **Measurement_Sensitivity** in *MalariaDiagnostic*.
- **event_name** – Description of the event.

- **node_ids** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of `NodeSetAll`.
- **positive_diagnosis_configs** – List of events to happen to an individual who receives a positive result from test.
- **negative_diagnosis_configs** – List of events to happen to individual who receives a negative result from test.
- **received_test_event** – String for individuals to broadcast upon receiving diagnostic.
- **IP_restrictions** – List of IndividualProperty key:value pairs that individuals must have to receive the diagnostic intervention. For example, [{"IndividualProperty1": "PropertyValue1"}, {"IndividualProperty2": "PropertyValue2"}]. Default is no restrictions.
- **NP_restrictions** – List of NodeProperty key:value pairs that nodes must have to receive the diagnostic intervention. For example, [{"NodeProperty1": "PropertyValue1"}, {"NodeProperty2": "PropertyValue2"}]. Default is no restrictions.
- **disqualifying_properties** – List of IndividualProperty key:value pairs that cause an intervention to be aborted. For example, [{"IndividualProperty1": "PropertyValue1"}, {"IndividualProperty2": "PropertyValue2"}].
- **trigger_condition_list** – List of events that will trigger a diagnostic survey.
- **listening_duration** – Duration after start day to stop listening for events, as specified in **trigger_condition_list**. Default is -1, non-stop listening.
- **triggered_campaign_delay** – Delay of running the intervention after receiving a trigger from the **trigger_condition_list**.
- **check_eligibility_at_trigger** – If triggered event is delayed, you have an option to check individual/node's eligibility at the initial trigger or when the event is actually distributed after delay.
- **expire_recent_drugs** – Adds [{"DrugStatus": None}] to **Property_Restrictions_Within_Node** for positive test config, so only those with that property receive drugs.

Returns None

emodpy_malaria.interventions.drug module

This module contains functionality for defining antimalarial drug interventions.

`emodpy_malaria.interventions.drug.AntimalarialDrug` (*campaign*, *start_day*: *int* = 1, *coverage*: *float* = 1.0, *drug_name*: *str* = 'Chloroquine', *node_ids*: *list* = *None*)

Add an antimalarial drug intervention to your campaign. This is equivalent to [AntimalarialDrug](#).

Parameters

- **campaign** – The `emod_api.campaign` object to which the intervention will be added.
- **start_day** – The day of the simulation on which the drug is distributed. We recommend aligning this with the start of the simulation.
- **coverage** – The proportion of the population that will receive the drug.

- **drug_name** – The name of the drug to distribute in a drug intervention. Possible values are contained in **Malaria_Drug_Params** in [Drugs and treatments](#). Use `set_team_drug_params()` to set those values.
- **node_ids** – The IDs of the nodes in which to distribute the drug.

Returns The intervention event.

`emodpy_malaria.interventions.drug.new_intervention_as_file(camp, start_day, filename=None)`

Take an [AntimalarialDrug](#) intervention from a JSON file and add it to your campaign.

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **start_day** – The day of the simulation on which the drug is distributed. We recommend aligning this with the start of the simulation.
- **filename** – The JSON file that contains the intervention.

Returns The filename.

emodpy_malaria.interventions.drug_campaign module

This module contains functionality for malaria intervention distribution via a cascade of care that may contain diagnostics and drug treatments.

`emodpy_malaria.interventions.drug_campaign.drug_configs_from_code(camp, drug_code: str = None)`

Add a single or multiple drug regimen to the configuration file based on its code and add the corresponding [AntimalarialDrug](#) intervention to the return dictionary. For example, passing the ALP drug code will add the drug configuration for artemether, lumefantrine, and primaquine to the configuration file and will return a dictionary containing a full treatment course for those three drugs. For more information, see **Malaria_Drug_Params** in [Drugs and treatments](#).

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **drug_code** – The code of the drug regimen. This must be listed in the `drug_cfg` dictionary.

Returns A dictionary containing the intervention for the given drug regimen.


```

emodpy_malaria.interventions.drug_campaign.add_drug_campaign(camp,      cam-
    campaign_type:
    str      = 'MDA',
    drug_code: str      =
    None,    start_days:
    list     = None, cov-
    erage:   float     = 1.0,
    repetitions: int     = 1,
    tsteps_btwn_repetitions:
    int      = 60, diag-
    nostic_type: str     =
    'BLOOD_SMEAR_PARASITES',
    diagnos-
    tic_threshold: float
    = 40,    measure-
    ment_sensitivity:
    float    = 0.1,
    fmda_radius:
    int      = 0,
    node_selection_type:
    str      = 'DIS-
    TANCE_ONLY',
    trigger_coverage:
    float    = 1.0, snow-
    balls:   int      = 0,
    treatment_delay:
    int      = 0,    trig-
    gered_campaign_delay:
    int      = 0,    node_ids:
    list     = None,
    target_group:
    any      = 'Everyone',
    drug_ineligibility_duration:
    int      = 0,
    node_property_restrictions:
    list     = None,
    ind_property_restrictions:
    list     = None,
    disqualify-
    ing_properties:
    list     = None,    trig-
    ger_condition_list:
    list     = None,    lis-
    tening_duration:
    int      = - 1,    adher-
    ent_drug_configs:
    list     = None,    tar-
    get_residents_only:
    int      = 1,
    check_eligibility_at_trigger:
    bool     =
    False,    receiv-
    ing_drugs_event_name='Received_Campaig

```

Add a drug intervention campaign from a list of malaria campaign types. This intervention uses the [Malari-](#)

a `Diagnostic` class to create either a scheduled or a triggered event to the campaign and the `AntimalarialDrug` class to configure drug interventions. You can also specify a delay period for a triggered event that broadcasts afterwards. If the campaign is repeated or triggered, separate `NodeLevelHealthTriggeredIV` interventions are created with a delay that sends an event to distribute drugs.

Parameters

- **camp** – The `emod_api.campaign` object to which the intervention will be added.
- **campaign_type** – The type of drug campaign. Available options are:
 - MDA** Add a mass drug administration intervention.
 - MSAT** Add a mass screening and treatment intervention.
 - SMC** Add a seasonal malaria chemoprevention intervention.
 - fMDA** Add a focal mass drug administration intervention based on results from a diagnostic survey, which is either scheduled or triggered (when **trigger_condition_list** is present).
 - MTAT** Add a mass testing and treatment intervention.
 - rfMSAT** Add a reactive focal mass screening and treatment intervention. Detecting malaria triggers diagnostic surveys to run on neighboring nodes and so on, up to the number of triggered interventions defined in the **snowballs** parameter.
 - rfMDA** Add a reactive focal mass drug administration intervention. This triggers `BroadcastEventToOtherNodes` to broadcast a “Give_Drugs_rfMDA” event, which triggers `MultiInterventionDistributor` to distribute drugs and a “Received_Treatment” event followed by a delayed “Give_Drugs_rfMDA” event to neighboring nodes, which will trigger another drug distribution.
- **drug_code** – The code of the drug regimen to distribute. This must be listed in the `drug_cfg` dictionary.
- **start_days** – List of start days (integers) when the drug regimen will be distributed. Due to diagnostic/treatment configuration, the earliest start day is 1. When **trigger_condition_list** is used, the first entry of **start_days** is the day to start listening for the trigger(s).
- **coverage** – The demographic coverage of the distribution (the fraction of people at home during the campaign).
- **repetitions** – The number of repetitions.
- **tsteps_btwn_repetitions** – The timesteps between the repetitions.
- **diagnostic_type** – The setting for **Diagnostic_Type** in `MalariaDiagnostic`. In addition to the accepted values listed there, you may also set `TRUE_INFECTION_STATUS`, which calls `StandardDiagnostic` instead.
- **diagnostic_threshold** – The setting for **Diagnostic_Threshold** in `MalariaDiagnostic`.
- **measurement_sensitivity** – The setting for **Measurement_Sensitivity** in `MalariaDiagnostic`.
- **detection_threshold** – The setting for **Detection_Threshold** in `MalariaDiagnostic`.
- **fmda_radius** – Radius (in km) of focal response upon finding infection. Used in simulations with many small nodes to simulate community health workers distributing drugs to surrounding houses. Used when **campaign_type** is set to `fMDA`.

- **node_selection_type** – The setting for **Node_Selection_Type** in [BroadcastEventToOtherNodes](#).
- **trigger_coverage** – The fraction of trigger events that will trigger reactive case detection (RCD). Used when **campaign_type** is set to rfMSAT or rfMDA. To set the fraction of individuals reached during RCD response, use **coverage**.
- **snowballs** – The number of times each triggered intervention will be distributed to surrounding nodes. For example, one snowball gives drugs to nodes neighboring the first node and two snowballs gives drugs to the nodes neighboring those nodes. Used when **campaign_type** is set to rfMSAT.
- **treatment_delay** – For **campaign_type** set to MSAT or fMDA, the length of time between administering a diagnostic and giving drugs; for values of rfMSAT or rfMDA, the length of time between treating the index case and triggering an RCD response.
- **triggered_campaign_delay** – When using **trigger_condition_list**, this indicates the delay period between receiving the trigger event and running the triggered campaign intervention.
- **node_ids** – The setting for **Node_List** in [Nodeset_Config](#) classes.
- **target_group** – A dictionary of {'agemin': x, 'agemax': y} to target MDA, SMC, MSAT, fMDA to individuals between x and y years of age. Default is Everyone.
- **drug_ineligibility_duration** – The number of days to set the **DrugStatus** individual property to **RecentDrug**, after which the property value is reverted. This property value prevents people from receiving drugs too frequently, but they can still receive diagnostics during this period. For more information, see [Targeting interventions to nodes or individuals](#).
- **node_property_restrictions** – The setting for **Node_Property_Restrictions** in [TriggeredEventCoordinator](#) that nodes must have to receive the diagnostic intervention.
- **ind_property_restrictions** – The setting for **Property_Restrictions_Within_Node** in [TriggeredEventCoordinator](#) that individuals must have to receive the diagnostic intervention.
- **disqualifying_properties** – The setting for **Disqualifying_Properties** in [AntimalarialDrug](#) or in [MalariaDiagnostic](#).
- **trigger_condition_list** – The setting for **Start_Trigger_Condition_List** in [TriggeredEventCoordinator](#).
- **listening_duration** – The setting for **Duration** in [TriggeredEventCoordinator](#).
- **adherent_drug_configs** – List of adherent drug configurations, which are dictionaries from `configure_adherent_drug`.
- **target_residents_only** – The setting for **Target_Residents_Only** in [TriggeredEventCoordinator](#).
- **check_eligibility_at_trigger** – Set to True to check the individual or node's eligibility at the initial trigger; set to False to check eligibility when the event is actually distributed after a delay.
- **receiving_drugs_event_name** – The event to broadcast when a person receives drugs.

Returns A dictionary with drug campaign parameters.

```
emodpy_malaria.interventions.drug_campaign.add_MDA(camp, start_days: list =
None, coverage: float =
1.0, drug_configs: list =
None, receiving_drugs_event:
emod_api.interventions.common.BroadcastEvent
= None, repetitions: int =
1, tsteps_btwn_repetitions:
int = 60, node_ids: list =
None, expire_recent_drugs:
emod_api.interventions.common.PropertyValueChanger
=
None,
node_property_restrictions: list =
None, ind_property_restrictions:
list = None, disqualify-
ing_properties: list = None,
target_group: any = 'Everyone',
trigger_condition_list: list =
None, listening_duration: int =
- 1, triggered_campaign_delay:
int = 0, target_residents_only: int
= 1, check_eligibility_at_trigger:
bool = False)
```

Add an MDA (mass drug administration) drug intervention to your campaign. See `add_drug_campaign()` for more information about each argument.

Returns None

```
emodpy_malaria.interventions.drug_campaign.add_MSAT(camp, start_days: list =
None, coverage: float =
1.0, drug_configs: list =
None, receiving_drugs_event:
emod_api.interventions.common.BroadcastEvent
= None, repetitions: int = 1,
tsteps_btwn_repetitions: int
= 60, treatment_delay: int
= 0, diagnostic_type: str =
'BLOOD_SMEAR_PARASITES',
diagnostic_threshold: float =
40, measurement_sensitivity:
float = 0.1, node_ids: list
= None, expire_recent_drugs:
emod_api.interventions.common.PropertyValueChanger
=
None,
node_property_restrictions:
list = None,
ind_property_restrictions: list =
None, disqualifying_properties:
list = None, target_group:
any = 'Everyone', trig-
ger_condition_list: list = None,
triggered_campaign_delay: int
= 0, listening_duration: int = -
1, check_eligibility_at_trigger:
bool = False)
```

Add an MSAT (mass screening and treatment) drug intervention to your campaign. See

`add_drug_campaign()` for more information about each argument.

Returns None

```
emodpy_malaria.interventions.drug_campaign.add_fmda(camp, start_days: list =
None, trigger_coverage:
float = 1, coverage: float
= 1, drug_configs: list =
None, receiving_drugs_event:
emod_api.interventions.common.BroadcastEvent
= None, repetitions: int = 1,
tsteps_btwn_repetitions: int
= 365, treatment_delay: int
= 0, diagnostic_type: str =
'BLOOD_SMEAR_PARASITES',
diagnostic_threshold: float = 40,
measurement_sensitivity: float
= 0.1, fmda_radius: int = 0,
node_selection_type: str = 'DIS-
TANCE_ONLY', node_ids: list
= None, expire_recent_drugs:
emod_api.interventions.common.PropertyValueChanger
=
None,
node_property_restrictions:
list = None,
ind_property_restrictions: list =
None, disqualifying_properties:
list = None, target_group:
any = 'Everyone', trig-
ger_condition_list: list = None,
listening_duration: int = - 1,
triggered_campaign_delay: int
= 0, check_eligibility_at_trigger:
bool = False)
```

Add an fMDA (focal mass drug administration) drug intervention to your campaign. See `add_drug_campaign()` for more information about each argument.

Returns None

```
emodpy_malaria.interventions.drug_campaign.add_rfMSAT(camp, start_day: int =
0, coverage: float = 1,
drug_configs: list = None,
receiving_drugs_event:
emod_api.interventions.common.BroadcastEvent
= None, listening_duration:
int = - 1, treatment_delay: int
= 0, trigger_coverage: float
= 1, diagnostic_type: str =
'BLOOD_SMEAR_PARASITES',
diagnostic_threshold: float =
40, measurement_sensitivity:
float = 0.1, fmda_radius: int
= 0, node_selection_type: str
= 'DISTANCE_ONLY', snow-
balls: int = 0, node_ids: list
= None, expire_recent_drugs:
emod_api.interventions.common.PropertyValueChang
=
None,
node_property_restrictions:
list = None,
ind_property_restrictions:
list = None, disqualify-
ing_properties: list = None)
```

Add a rfMSAT (reactive focal mass screening and treatment) drug intervention to your campaign. See [add_drug_campaign\(\)](#) for more information about each argument.

Returns None

```
emodpy_malaria.interventions.drug_campaign.add_rfMDA(camp, start_day: int =
0, coverage: float = 1,
drug_configs: list = None,
receiving_drugs_event:
emod_api.interventions.common.BroadcastEvent
= None, listening_duration:
int = - 1, treatment_delay:
int = 0, trigger_coverage:
float = 1, fmda_radius: int
= 0, node_selection_type:
str = 'DISTANCE_ONLY',
node_ids: list = None,
expire_recent_drugs:
emod_api.interventions.common.PropertyValueChanger
=
None,
node_property_restrictions:
list = None,
ind_property_restrictions:
list = None, disqualify-
ing_properties: list = None)
```

Add an rfMDA (reactive focal mass drug administration) drug intervention to your campaign. See [add_drug_campaign\(\)](#) for more information about each argument.

Returns None

```
emodpy_malaria.interventions.drug_campaign.fmda_cfg(camp, fmda_type: any = 0,
                                                    node_selection_type: str = 'DIS-
TANCE_ONLY', event_trigger:
                                                    str = 'Give_Drugs')
```

Create an fMDA (focal mass drug administration) configuration.

Parameters

- **fmda_type** – The setting for **Max_Distance_To_Other_Nodes_Km** in [BroadcastEventToOtherNodes](#).
- **node_selection_type** – The setting for **Node_Selection_Type** in [BroadcastEventToOtherNodes](#).
- **event_trigger** – The setting for **Event_Trigger** in [BroadcastEventToOtherNodes](#).

Returns Configured [BroadcastEventToOtherNodes](#) intervention.

emodpy_malaria.interventions.inputeir module

```
emodpy_malaria.interventions.inputeir.new_intervention(campaign, monthly_eir: list
                                                         = None, daily_eir: list
                                                         = None, age_dependence:
                                                         str = 'OFF', scaling_factor:
                                                         float = 1.0)
```

Create the InputEIR intervention itself that will be nestled inside an event coordinator.

Parameters

- **campaign** – Passed in campaign (from `emod_api.campaign`)
- **monthly_eir** – An array of 12 elements that contain an entomological inoculation rate (EIR) for each month; Each value should be between 0 and 1000
- **daily_eir** – An array of 365 values where each value is the mean number of infectious bites experienced by an individual for that day of the year
- **age_dependence** – Determines how InputEIR depends on the age of the target. Options are “OFF”, “LINEAR”, “SURFACE_AREA_DEPENDENT”
- **scaling_factor** – A modifier that is multiplied by the EIR determined for the current day

Returns InputEIR intervention

```
emodpy_malaria.interventions.inputeir.InputEIR(campaign, monthly_eir: list = None,
                                                  daily_eir: list = None, start_day:
                                                  int = 1, node_ids: list = None,
                                                  age_dependence: str = 'OFF', scal-
                                                  ing_factor: float = 1.0)
```

Create a full CampaignEvent that distributes InputEIR to a population.

Parameters

- **campaign** – Passed in campaign (from `emod_api.campaign`)
- **monthly_eir** – An array of 12 elements that contain an entomological inoculation rate (EIR) for each month; Each value should be between 0 and 1000
- **daily_eir** – An array of 365 values where each value is the mean number of infectious bites experienced by an individual for that day of the year

- **start_day** – The day on which the monthly_eir cycle starts
- **node_ids** – Nodes to which this intervention is applied
- **age_dependence** – Determines how InputEIR depends on the age of the target. Options are “OFF”, “LINEAR”, “SURFACE_AREA_DEPENDENT”
- **scaling_factor** – A modifier that is multiplied by the EIR determined for the current day

Returns Campaign event to be added to campaign (from `emod_api.campaign`)

```
emodpy_malaria.interventions.inputeir.new_intervention_as_file(campaign,  
                                                                start_day: int =  
                                                                0, monthly_eir:  
                                                                list = None,  
                                                                daily_eir: list =  
                                                                None, filename:  
                                                                str = None)
```

Create an InputEIR intervention as its own file.

Parameters

- **campaign** – Passed in campaign (from `emod_api.campaign`)
- **start_day** – The day on which the monthly_eir cycle starts
- **monthly_eir** – An array of 12 elements that contain an entomological inoculation rate (EIR) for each month; Each value should be between 0 and 1000
- **daily_eir** – An array of 365 values where each value is the mean number of infectious bites experienced by an individual for that day of the year
- **filename** – filename used for the file created

Returns The filename of the file created

```
emodpy_malaria.interventions.inputeir.add_InputEIR(campaign, monthly_eir: list =  
                                                    None, daily_eir: list = None,  
                                                    start_day: int = 1, node_ids:  
                                                    list = None, age_dependence: str  
                                                    = 'OFF', scaling_factor: float =  
                                                    1.0)
```

Wrapper that creates a full CampaignEvent that distributes InputEIR to a population AND adds it to the campaign.

Parameters

- **campaign** – Passed in campaign (from `emod_api.campaign`)
- **monthly_eir** – An array of 12 elements that contain an entomological inoculation rate (EIR) for each month; Each value should be between 0 and 1000
- **daily_eir** – An array of 365 values where each value is the mean number of infectious bites experienced by an individual for that day of the year
- **start_day** – The day on which the monthly_eir cycle starts
- **node_ids** – Nodes to which this intervention is applied
- **age_dependence** – Determines how InputEIR depends on the age of the target. Options are “OFF”, “LINEAR”, “SURFACE_AREA_DEPENDENT”
- **scaling_factor** – A modifier that is multiplied by the EIR determined for the current day

emodpy_malaria.interventions.irs module

```
emodpy_malaria.interventions.irs.add_irs_housing_modification(campaign,
                                                                start_day: int
                                                                = 1, coverage: float = 1,
                                                                killing_initial_effect: float = 1,
                                                                killing_box_duration_days: int = 0,
                                                                killing_exponential_decay_constant_days: int = 90,
                                                                repelling_initial_effect: float = 0,
                                                                repelling_box_duration_days: int = 0,
                                                                repelling_exponential_decay_constant_days: int = 90,
                                                                insecticide: str = None,
                                                                node_ids: list = None)
```

Adds IRSHousingModification intervention to the campaign. The IRSHousingModification intervention class includes Indoor Residual Spraying (IRS) in the simulation. IRS is another key vector control tool in which insecticide is sprayed on the interior walls of a house so that mosquitoes resting on the walls after consuming a blood meal will die. IRS can also have a repellent effect. Because this class is distributed to individuals, it can target subgroups of the population. To target all individuals in a node, use IndoorSpaceSpraying. Do not use IRSHousingModification and IndoorSpaceSpraying together.

Parameters

- **campaign** – A campaign builder that also contains schema_path parameters
- **start_day** – The day on which the intervention is distributed
- **coverage** – The fraction of individuals in the target demographic that will receive this intervention
- **killing_initial_effect** – Initial strength of the Killing effect. The effect decays over time.
- **killing_box_duration_days** – Box duration of effect in days before the decay of Killing Initial Effect.
- **killing_exponential_decay_constant_days** – The exponential decay length, in days of the Killing Initial Effect.
- **repelling_initial_effect** – Initial strength of the Killing effect. The effect decays over time.
- **repelling_box_duration_days** – Box duration of effect in days before the decay of Repelling Initial Effect.
- **repelling_exponential_decay_constant_days** – The exponential decay length, in days of the Repelling Initial Effect.
- **insecticide** – The name of the insecticide defined in config.Insecticides for this intervention. If insecticides are being used, then this must be defined as one of those values. If

they are not being used, then this does not need to be specified or can be empty string. It cannot have a value if `config.Insecticides` does not define anything.

- **node_ids** – A list of node ids to which this intervention will be distributed. None or [] distributes intervention to all nodes.

Returns Nothing

```
emodpy_malaria.interventions.irs.new_intervention_as_file(campaign, start_day,
                                                         filename=None)
```

emodpy_malaria.interventions.ivermectin module

```
emodpy_malaria.interventions.ivermectin.Ivermectin(schema_path_container,
                                                    start_day=1, killing_initial_effect:
                                                    float = 1, demographic_coverage: float = 1.0, target_num_individuals: int = None,
                                                    killing_box_duration: int = 0,
                                                    killing_exponential_decay_rate:
                                                    float = 0)
```

Create a scheduled Ivermectin CampaignEvent which can then be added to a campaign.

Parameters

- **schema_path_container** – schema path container? a way to pass the schema to the python script
- **start_day** – day to give out this intervention
- **demographic_coverage** – probability of choosing an individual, is ignored if “target_num_individuals” is set
- **target_num_individuals** – number of individuals to receive ivermectin, demographic_coverage will be ignored if this is set
- **killing_initial_effect** – initial parasite killing effect
- **killing_box_duration** – box duration for killing effect
- **killing_exponential_decay_rate** – rate at which killing effect decays per day. Use 0 for box duration only.

Returns CampaignEvent which then can be added to the campaign file

emodpy_malaria.interventions.mosquitorelease module

```
emodpy_malaria.interventions.mosquitorelease.MosquitoRelease(campaign,
                                                                start_day: int = 0,
                                                                released_number:
                                                                int = None, re-
                                                                leased_fraction:
                                                                float = None, re-
                                                                leased_infectious:
                                                                float = 0, re-
                                                                leased_species:
                                                                str = 'arabiensis',
                                                                released_genome:
                                                                list = None,
                                                                node_ids: list =
                                                                None)
```

Release mosquitoes of a given species and genome into each node.

Parameters

- **campaign** –
- **start_day** – The day to release the vectors.
- **released_number** – The number of vectors to release, sets Released_Type = “FIXED_NUMBER”
- **released_fraction** – The fraction of the current population of mosquitoes to release. The ‘population’ will depend on the gender of the mosquitoes being released and it will be the population from the previous time step. Sets Released_Type = “FRACTION”
- **released_infectious** – The fraction of vectors released that are to be infectious. One can only use this feature when ‘Malaria_Model’!=‘MALARIA_MECHANISTIC_MODEL_WITH_PARASITE_GENETICS’
- **released_species** – The case sensitive name of the species of vectors to be released.
- **released_genome** – This defines the alleles of the genome of the vectors to be released. It must define all of the alleles including the gender ‘gene’. ‘*’ is not allowed.
- **node_ids** – The list of node IDs to receive a release of vectors. The same number of vectors will be released to each node.

Returns: Formatted intervention

```
emodpy_malaria.interventions.mosquitorelease.new_intervention_as_file(camp,
                                                                start_day:
                                                                int
                                                                = 1,
                                                                file-
                                                                name:
                                                                str =
                                                                None)
```

Parameters

- **camp** –
- **start_day** – The day to release the vectors.
- **filename** – name of campaign filename to be created

Returns returns filename

emodpy_malaria.interventions.outbreak module

```
emodpy_malaria.interventions.outbreak.add_outbreak_individual(campaign,
                                                                start_day: int
                                                                = 1, demo-
                                                                graphic_coverage:
                                                                float = 1.0, tar-
                                                                get_num_individuals:
                                                                int = None,
                                                                node_ids: list
                                                                = None, repeti-
                                                                tions: int = 1,
                                                                timesteps_between_repetitions:
                                                                int = 365,
                                                                ind_property_restrictions:
                                                                list = None,
                                                                node_property_restrictions:
                                                                list = None, tar-
                                                                get_age_min:
                                                                int = 0, tar-
                                                                get_age_max:
                                                                int = 125, tar-
                                                                get_gender:
                                                                str = 'All', ig-
                                                                nore_immunity:
                                                                bool = True,
                                                                incuba-
                                                                tion_period_override:
                                                                int = -1, antigen:
                                                                int = 0, genome:
                                                                int = 0, broad-
                                                                cast_event: str =
                                                                None)
```

Adds a scheduled OutbreakIndividual intervention. This is set up to be used with Malaria-Ongoing branch.

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – The day the intervention is given out.
- **demographic_coverage** – This value is the probability that each individual in the target population will receive the intervention. It does not guarantee that the exact fraction of the target population set by Demographic_Coverage receives the intervention.
- **target_num_individuals** – The exact number of people to select out of the targeted group. If this value is set, demographic_coverage parameter is ignored
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **repetitions** – The number of times an intervention is given, used with timesteps_between_repetitions. -1 means the intervention repeats forever. Sets **Number_Repetitions**

- **timesteps_between_repetitions** – The interval, in timesteps, between repetitions. Ignored if repetitions = 1. Sets **Timesteps_Between_Repetitions**
- **ind_property_restrictions** – A list of dictionaries of IndividualProperties, which are needed for the individual to receive the intervention. Sets the **Property_Restrictions_Within_Node**
- **node_property_restrictions** – A list of the NodeProperty key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**
- **target_age_min** – The lower end of ages targeted for an intervention, in years. Sets **Target_Age_Min**
- **target_age_max** – The upper end of ages targeted for an intervention, in years. Sets **Target_Age_Max**
- **target_gender** – The gender targeted for an intervention: All, Male, or Female.
- **ignore_immunity** – Individuals will be force-infected (with a specific strain) regardless of actual immunity level when set to True (1). Default is True (1). The person will or will not get an infection based on their immunity level if this is set to False.
- **incubation_period_override** – The incubation period, in days, that infected individuals will go through before becoming infectious. This value overrides the incubation period set in the configuration file. Set to -1 to honor the configuration parameter settings
- **antigen** – The antigenic base strain ID of the outbreak infection
- **genome** – The genetic substrain ID of the outbreak infection
- **broadcast_event** – Optional event that will be sent out at the same time as outbreak is distributed

Returns Nothing

```

emodpy_malaria.interventions.outbreak.add_outbreak_malaria_genetics(campaign,
                                                                    start_day:
                                                                    int = 1,
                                                                    demo-
                                                                    graphic_coverage:
                                                                    float =
                                                                    1.0, tar-
                                                                    get_num_individuals:
                                                                    int =
                                                                    None,
                                                                    node_ids:
                                                                    list =
                                                                    None,
                                                                    repeti-
                                                                    tions:
                                                                    int = 1,
                                                                    timesteps_between_repetitions:
                                                                    int
                                                                    = 365,
                                                                    ind_property_restrictions:
                                                                    list =
                                                                    None,
                                                                    node_property_restrictions:
                                                                    list =
                                                                    None,
                                                                    tar-
                                                                    get_age_min:
                                                                    int =
                                                                    0, tar-
                                                                    get_age_max:
                                                                    int =
                                                                    125, tar-
                                                                    get_gender:
                                                                    str =
                                                                    'All', ig-
                                                                    nore_immunity:
                                                                    bool =
                                                                    True,
                                                                    incuba-
                                                                    tion_period_override:
                                                                    int = -
                                                                    1, cre-
                                                                    ate_nucleotide_sequence_from:
                                                                    str =
                                                                    'BAR-
                                                                    CODE_STRING',
                                                                    bar-
                                                                    code_string:
                                                                    str =
                                                                    None,
                                                                    drug_resistant_string:
                                                                    str =
                                                                    None,
                                                                    msp_variant_value:
                                                                    int =
                                                                    None,
                                                                    pfp-
                                                                    mp_values:
                                                                    list =
                                                                    None,
                                                                    bar-

```

Creates a scheduled OutbreakIndividualMalariaGenetics CampaignEvent which can then be added to a campaign.

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – The day the intervention is given out.
- **demographic_coverage** – This value is the probability that each individual in the target population will receive the intervention. It does not guarantee that the exact fraction of the target population set by Demographic_Coverage receives the intervention.
- **target_num_individuals** – The exact number of people to select out of the targeted group. If this value is set, demographic_coverage parameter is ignored.
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **repetitions** – The number of times an intervention is given, used with timesteps_between_repetitions. -1 means the intervention repeats forever. Sets **Number_Repetitions**
- **timesteps_between_repetitions** – The interval, in timesteps, between repetitions. Ignored if repetitions = 1. Sets **Timesteps_Between_Repetitions**
- **ind_property_restrictions** – A list of dictionaries of IndividualProperties, which are needed for the individual to receive the intervention. Sets the **Property_Restrictions_Within_Node**
- **node_property_restrictions** – A list of the NodeProperty key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**
- **target_age_min** – The lower end of ages targeted for an intervention, in years. Sets **Target_Age_Min**
- **target_age_max** – The upper end of ages targeted for an intervention, in years. Sets **Target_Age_Max**
- **target_gender** – The gender targeted for an intervention: All, Male, or Female.
- **ignore_immunity** – Individuals will be force-infected (with a specific strain) regardless of actual immunity level when set to True (1). Default is True (1). The person will or will not get an infection based on their immunity level if this is set to False.
- **incubation_period_override** – The incubation period, in days, that infected individuals will go through before becoming infectious. This value overrides the incubation period set in the configuration file. Set to -1 to honor the configuration parameter settings
- **create_nucleotide_sequence_from** – A string that indicates how the genomes are created. Possible values are: BARCODE_STRING, ALLELE_FREQUENCIES, NUCLEOTIDE_SEQUENCE.
- **barcode_string** – A series of nucleotide base letters (A, C, G, T) that represent the values at locations in the genome. The length of the string depends on the number of locations defined in config.Parasite_Genetics.Barcode_Genome_Locations. Each character of the string corresponds to one of the locations. The locations are assumed to be in ascending order. Also depends on create_nucleotide_sequence_from when it is equal to NUCLEOTIDE_SEQUENCE or BARCODE_STRING.

- **drug_resistant_string** – A series of nucleotide base letters (A, C, G, T) that represent the values at locations in the genome. The length of the string depends on the number of locations defined in `config.Parasite_Genetics.Drug_Resistant_Genome_Locations`. Each character of the string corresponds to one of the locations. The locations are assumed to be in ascending order. Also depends on `create_nucleotide_sequence_from` when it is equal to `NUCLEOTIDE_SEQUENCE` or `BARCODE_STRING`.
- **mvp_variant_value** – The Merozoite Surface Protein value used to determine how the antibodies recognize the merozoites. This value depends on `config.Falciparum_MSP_Variants` and must be less than or equal to it. It also depends on `create_nucleotide_sequence_from` when it is equal to `NUCLEOTIDE_SEQUENCE`.
- **pfemp1_variants_values** – The PfEMP1 Variant values / major epitopes used to define how the antibodies recognize the infected red blood cells. The values of the array depend on `config.Falciparum_PfEMP1_Variants` and must be less than or equal to it. There must be exactly 50 values – one for each epitope. It also depends on `create_nucleotide_sequence_from` when it is equal to `NUCLEOTIDE_SEQUENCE`.
- **barcode_allele_frequencies_per_genome_location** – The fractions of allele occurrences for each location in the barcode. This 2D array should have one array for each location/character in the barcode. For each location, there should be four values between 0 and 1 indicating the probability that specific character appears. The possible letters are: A=0, C=1, G=2, T=3. It also depends on `create_nucleotide_sequence_from` when it is equal to `ALLELE_FREQUENCIES`. The frequencies should sum up to 1.
- **drug_resistant_allele_frequencies_per_genome_location** – The fractions of allele occurrences for each location in the drug resistant markers. This 2D array should have one array for each drug resistant location. For each location, there should be four values between 0 and 1 indicating the probability that specific character will appear. The possible letters are 'A'=0, 'C'=1, 'G'=2, 'T'=3. It also depends on `create_nucleotide_sequence_from` when it is equal to `ALLELE_FREQUENCIES`. The frequencies should sum up to 1.

Returns CampaignEvent which then can be added to the campaign file

```

emodpy_malaria.interventions.outbreak.add_outbreak_malaria_var_genes(campaign,
                                                                    start_day:
                                                                    int = 1,
                                                                    demo-
                                                                    graphic_coverage:
                                                                    float
                                                                    = 1.0,
                                                                    tar-
                                                                    get_num_individuals:
                                                                    int =
                                                                    None,
                                                                    node_ids:
                                                                    list =
                                                                    None,
                                                                    repeti-
                                                                    tions:
                                                                    int = 1,
                                                                    timesteps_between_repetitions:
                                                                    int =
                                                                    365,
                                                                    ind_property_restrictions:
                                                                    list =
                                                                    None,
                                                                    node_property_restrictions:
                                                                    list =
                                                                    None,
                                                                    tar-
                                                                    get_age_min:
                                                                    int =
                                                                    0, tar-
                                                                    get_age_max:
                                                                    int =
                                                                    125,
                                                                    tar-
                                                                    get_gender:
                                                                    str =
                                                                    'All',
                                                                    ig-
                                                                    nore_immunity:
                                                                    bool =
                                                                    True,
                                                                    incuba-
                                                                    tion_period_override:
                                                                    int =
                                                                    - 1,
                                                                    irbc_type:
                                                                    list =
                                                                    None,
                                                                    mi-
                                                                    nor_epitope_type:
                                                                    list =
                                                                    None,
                                                                    msh_type:
                                                                    int =
                                                                    None)

```

Creates a scheduled OutbreakIndividualMalariaGenetics CampaignEvent which can then be added to a campaign.

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – The day the intervention is given out.
- **demographic_coverage** – This value is the probability that each individual in the target population will receive the intervention. It does not guarantee that the exact fraction of the target population set by Demographic_Coverage receives the intervention.
- **target_num_individuals** – The exact number of people to select out of the targeted group. If this value is set, demographic_coverage parameter is ignored
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **repetitions** – The number of times an intervention is given, used with timesteps_between_repetitions. -1 means the intervention repeats forever. Sets **Number_Repetitions**
- **timesteps_between_repetitions** – The interval, in timesteps, between repetitions. Ignored if repetitions = 1. Sets **Timesteps_Between_Repetitions**
- **ind_property_restrictions** – A list of dictionaries of IndividualProperties, which are needed for the individual to receive the intervention. Sets the **Property_Restrictions_Within_Node**
- **node_property_restrictions** – A list of the NodeProperty key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**
- **target_age_min** – The lower end of ages targeted for an intervention, in years. Sets **Target_Age_Min**
- **target_age_max** – The upper end of ages targeted for an intervention, in years. Sets **Target_Age_Max**
- **target_gender** – The gender targeted for an intervention: All, Male, or Female.
- **ignore_immunity** – Individuals will be force-infected (with a specific strain) regardless of actual immunity level when set to True (1). Default is True (1). The person will or will not get an infection based on their immunity level if this is set to False.
- **incubation_period_override** – The incubation period, in days, that infected individuals will go through before becoming infectious. This value overrides the incubation period set in the configuration file. Set to -1 to honor the configuration parameter settings
- **irbc_type** – The array PfEMP1 Major epitope variant values. There must be exactly 50 values. Min value = 0, MAX value = config.Falciparum_PfEMP1_Variants.
- **minor_epitope_type** – The array PfEMP1 Minor epitope variant values. There must be exactly 50 values. Min value = 0, MAX value = config.Falciparum_Nonspecific_Types * MINOR_EPITOPE_VARS_PER_SET(=5) .
- **mssp_type** – The Merozoite Surface Protein variant value of this infection. Min value = 0, MAX value = config.Falciparum_MSP_Variants.

Returns CampaignEvent which then can be added to the campaign file

```

emodpy_malaria.interventions.outbreak.add_campaign_event (campaign, start_day:
                                                         int = 1, demo-
                                                         graphic_coverage:
                                                         float = 1.0, tar-
                                                         get_num_individuals:
                                                         int = None, node_ids:
                                                         list = None, repe-
                                                         titions: int = 1,
                                                         timesteps_between_repetitions:
                                                         int = 365,
                                                         ind_property_restrictions:
                                                         list = None,
                                                         node_property_restrictions:
                                                         list = None, tar-
                                                         get_age_min: int =
                                                         0, target_age_max: int =
                                                         125, target_gender: str =
                                                         'All', intervention: any =
                                                         None)

```

Adds a campaign event to the campaign with a passed in intervention.

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – The day the intervention is given out.
- **demographic_coverage** – This value is the probability that each individual in the target population will receive the intervention. It does not guarantee that the exact fraction of the target population set by Demographic_Coverage receives the intervention.
- **target_num_individuals** – The exact number of people to select out of the targeted group. If this value is set, demographic_coverage parameter is ignored
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **repetitions** – The number of times an intervention is given, used with timesteps_between_repetitions. -1 means the intervention repeats forever. Sets **Number_Repetitions**
- **timesteps_between_repetitions** – The interval, in timesteps, between repetitions. Ignored if repetitions = 1. Sets **Timesteps_Between_Repetitions**
- **ind_property_restrictions** – A list of dictionaries of IndividualProperties, which are needed for the individual to receive the intervention. Sets the **Property_Restrictions_Within_Node**
- **node_property_restrictions** – A list of the NodeProperty key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**
- **target_age_min** – The lower end of ages targeted for an intervention, in years. Sets **Target_Age_Min**
- **target_age_max** – The upper end of ages targeted for an intervention, in years. Sets **Target_Age_Max**
- **target_gender** – The gender targeted for an intervention: All, Male, or Female.

- **intervention** – Intervention or a list of interventions to be distributed by this event

Returns:

emodpy_malaria.interventions.outdoorrestkill module

```
emodpy_malaria.interventions.outdoorrestkill.add_OutdoorRestKill(campaign,  
                                                                    start_day: int  
                                                                    = 1, insecti-  
                                                                    cide_name:  
                                                                    str = None,  
                                                                    kill  
                                                                    ing_initial_effect:  
                                                                    float = 1,  
                                                                    kill  
                                                                    ing_box_duration:  
                                                                    int = 365,  
                                                                    kill  
                                                                    ing_exponential_decay_rate:  
                                                                    float = 0,  
                                                                    node_ids: list  
                                                                    = None)
```

Adds an OutdoorRestKill intervention to the campaign

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – the day on which to distribute the intervention
- **insecticide_name** – Name of the insecticide
- **kill**
ing_initial_effect – **Initial_Effect** in the *Killing_Config**
- **kill**
ing_box_duration – Length in days before the **Initial_Effect** starts to decay
- **kill**
ing_exponential_decay_rate – The rate of decay of the *Initial_Effect**
- **node_ids** – List of nodes to which to distribute the intervention. None or empty list implies “all nodes”.

Returns configured campaign object

emodpy_malaria.interventions.scale_larval_habitats module

emodpy_malaria.interventions.scale_larval_habitats.add_scale_larval_habitats(*campaign*,
df=None,
start_day:
int
= 0,
rep-
e-
ti-
tions:
int
= 1,
timesteps_between:
int
= 365,
node_property_restrictions:
list
= None)

Reduce available larval habitat in a node-specific way.

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **df** – The dataframe containing habitat scale factors. **Examples:**

```
Scale TEMPORARY_RAINFALL by 3-fold for all nodes, all species:
df = pd.DataFrame({ 'TEMPORARY_RAINFALL': [3]})

Scale TEMPORARY_RAINFALL by 3-fold for all nodes, arabiensis only:
df = pd.DataFrame({ 'TEMPORARY_RAINFALL.arabiensis': [3]})

Scale differently by node ID:
df = pd.DataFrame({ 'NodeID' : [0, 1, 2, 3, 4],
                    'CONSTANT': [1, 0, 1, 1, 1],
                    'TEMPORARY_RAINFALL': [1, 1, 0, 1, 0]})

Scale differently by both node ID and species:
df = pd.DataFrame({ 'NodeID' : [0, 1, 2, 3, 4],
                    'CONSTANT.arabiensis': [1, 0, 1, 1, 1],
                    'TEMPORARY_RAINFALL.arabiensis': [1, 1, 0, 1, 0],
                    'CONSTANT.funestus': [1, 0, 1, 1, 1]})

Scale some habitats by species and others same for all species:
df = pd.DataFrame({ 'NodeID' : [0, 1, 2, 3, 4],
                    'CONSTANT.arabiensis': [1, 0, 1, 1, 1],
                    'TEMPORARY_RAINFALL.arabiensis': [1, 1, 0, 1, 0],
                    'CONSTANT.funestus': [1, 0, 1, 1, 1],
                    'LINEAR_SPLINE': [1, 1, 0, 1, 0]})
```

(continues on next page)

(continued from previous page)

```
Scale nodes at different dates:
df = pd.DataFrame({ 'NodeID' : [0, 1, 2, 3, 4],
                    'CONSTANT': [1, 0, 1, 1, 1],
                    'TEMPORARY_RAINFALL': [1, 1, 0, 1, 0],
                    'Start_Day': [0, 30, 60, 65, 65]
                    })
```

- **start_day** – The date that habitats are scaled for all scaling actions specified in **df**. Used only if there is no **Start_Day** column in **df**.
- **repetitions** – The number of times to repeat the intervention.
- **timesteps_between_repetitions** – The number of time steps between repetitions.
- **node_property_restrictions** – The node property values to target; used with **NodePropertyRestrictions**. For example, “[{ “NodeProperty1” : “PropertyValue1” }, { “NodeProperty2”: “PropertyValue2”}, ...]”.

Returns None

```
emodpy_malaria.interventions.scale_larval_habitats.add_habitat_reduction_event (campaign,
start_day:
int,
node_ids:
list,
habitat_scales:
list,
repetitions:
int,
timesteps_between_repetitions:
int,
node_property_restrictions:
list)
```

Add a campaign event to reduce vector’s larval habitat(s).

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – The day the intervention is given out.
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **habitat_scales** – List of dictionaries for scaling larval habitats. Examples:

```
[{"Habitat": "ALL_HABITATS", "Species": "ALL_SPECIES", "Factor": 0.5},
{"Habitat": "CONSTANT", "Species": "arabiensis", "Factor": 2}]
```

- **repetitions** – The number of times an intervention is given, used with `timesteps_between_repetitions`. -1 means the intervention repeats forever. Sets **Number_Repetitions**
- **timesteps_between_repetitions** – The interval, in timesteps, between repetitions. Ignored if `repetitions = 1`. Sets **Timesteps_Between_Repetitions**
- **node_property_restrictions** – A list of the NodeProperty key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**

Returns Nothing

emodpy_malaria.interventions.spacespraying module

```
emodpy_malaria.interventions.spacespraying.SpaceSpraying(campaign, start_day: int
                                                           = 1, spray_coverage:
                                                           float = 1.0, killing_effect:
                                                           float = 1, insecticide: str
                                                           = None, box_duration:
                                                           int = 0, decay_rate: float
                                                           = 0, node_ids: list =
                                                           None)
```

Create a new SpaceSpraying scheduled campaign intervention. Note: for WaningEffect, Decay_Time_Constant = 1.0/decay_rate box_duration = 0 + decay_rate > 0 => WaningEffectExponential box_duration > 0 + decay_rate = 0 => WaningEffectBox/Constant (depending on duration) box_duration > 0 + decay_rate > 0 => WaningEffectBoxExponential

Args: campaign: start_day: the day to distribute the SpaceSpraying intervention spray_coverage: how much of each node to cover (total portion killed = killing effect * coverage) killing_effect: portion of vectors killed by the intervention (Initial_Effect in WaningEffect) insecticide: insecticide, if used box_duration: Box_Duration of the WaningEffect decay_rate: decay_rate of the WaningEffect, gets set as Decay_Time_Constant = 1.0/decay_rate node_ids: list of node ids to which distribute the intervention

Returns: The formatted intervention ready to be added to the campaign.

```
emodpy_malaria.interventions.spacespraying.new_intervention_as_file(campaign,
                                                                      start_day:
                                                                      int = 0,
                                                                      file-
                                                                      name:
                                                                      str =
                                                                      None)
```

Creates a file with SpaceSpray intervention :param campaign: :param start_day: the day to distribute the SpaceSpraying intervention :param filename: name of the filename created

Returns: filename of the file created

emodpy_malaria.interventions.sugartrap module

```
emodpy_malaria.interventions.sugartrap.SugarTrap(campaign, start_day: int = 0,
                                                    killing_effect: float = 1, insecti-
                                                    cide: str = None, box_duration:
                                                    int = 100, decay_rate: float =
                                                    0, expiration_constant: int = 100,
                                                    node_ids=None)
```

Create a new SugarTrap scheduled campaign intervention. Note: for WaningEffect, Decay_Time_Constant = 1.0/decay_rate. box_duration = 0 + decay_rate > 0 => WaningEffectExponential box_duration > 0 + decay_rate = 0 => WaningEffectBox/Constant (depending on duration) box_duration > 0 + decay_rate > 0 => WaningEffectBoxExponential

Parameters

- **campaign** – campaign builder.
- **start_day** – the day to distribute the SpaceSpraying intervention
- **killing_effect** – portion of vectors killed by the intervention (Initial_Effect in WaningEffect)
- **insecticide** – insecticide, if used
- **box_duration** – Box_Duration of the WaningEffect
- **decay_rate** – decay_rate of the WaningEffect, gets set as Decay_Time_Constant = 1.0/decay_rate
- **node_ids** – list of node ids to which distribute the intervention
- **expiration_constant** – how long after distribution of intervention it gets discarded

Returns The formatted intervention ready to be added to the campaign.

```
emodpy_malaria.interventions.sugartrap.new_intervention_as_file(campaign,
                                                                  start_day: int
                                                                  = 0, filename:
                                                                  str = None)
```

Create new campaign file with a single event which distributes a SugarTrap intervention mostly with defaults. Useful for sanity testing and first time users. :param campaign: campaign builder. :param start_day: the day to distribute the SpaceSpraying intervention :param filename: name of the filename created

Returns Filename of the file created.

emodpy_malaria.interventions.treatment_seeking module

```
emodpy_malaria.interventions.treatment_seeking.add_treatment_seeking(camp,
                                                                        start_day:
                                                                        int = 1,
                                                                        targets:
                                                                        list =
                                                                        None,
                                                                        drug:
                                                                        list =
                                                                        None,
                                                                        node_ids:
                                                                        list =
                                                                        None,
                                                                        ind_property_restrictions:
                                                                        list =
                                                                        None,
                                                                        drug_ineligibility_duration:
                                                                        float =
                                                                        0, du-
                                                                        ration:
                                                                        int =
                                                                        - 1,
                                                                        broad-
                                                                        cast_event_name:
                                                                        str =
                                                                        'Re-
                                                                        ceived_Treatment')
```

Add an event-triggered drug-seeking behavior intervention to the campaign using the **NodeLevelHealthTriggeredIV**. The intervention will distribute drugs to targeted individuals within the node.

Parameters

- **camp** – object for building, modifying, and writing campaign configuration files.
- **start_day** – Start day of intervention.
- **targets** – List of dictionaries defining the trigger event and coverage for and
- **of individuals to target with the intervention. Default is**
(properties) – [{ "trigger": "NewClinicalCase", "coverage": 0.8, "agemin": 15, "agemax": 70, "seek": 0.4, "rate": 0.3},
 {"trigger": "NewSevereCase", "coverage": 0.8, "seek": 0.6, "rate": 0.5}].
- **drug** – List of drug(s) to administer. Default is ["Artemether", "Lumefantrine"].
- **node_ids** – The list of nodes to apply this intervention to (**Node_List**
- **If not provided, set value of NodeSetAll. (parameter)**–
- **ind_property_restrictions** – List of IndividualProperty key:value pairs that
- **must have to receive the intervention. For example,**
(individuals) – ["IndividualProperty1:PropertyValue1",
 "IndividualProperty2:PropertyValue2"].
- **drug_ineligibility_duration** – number of days for which an individual will be ineligible for more drugs

- **duration** – duration from start_day until people will no longer seek drugs when sick. Default is -1, where this never happens.
- **broadcast_event_name** – Event to broadcast when successful health seeking behavior.
- **is Received_Treatment.** (Default) –

Returns None

emodpy_malaria.interventions.udbednet module

```
emodpy_malaria.interventions.udbednet.UDBednet(camp, start_day: int = 1, discard_config: dict = None, coverage: float = 1, ind_property_restrictions: list = None, blocking_eff: float = 0.9, blocking_constant_duration: int = 0, blocking_decay_rate: float = 0.0013698630136986301, killing_eff: float = 0.6, killing_constant_duration: int = 0, killing_decay_rate: float = 0.0006849315068493151, repelling_eff: float = 0, repelling_constant_duration: int = 0, repelling_decay_rate: float = 0.0006849315068493151, iv_name: str = 'UsageDependentBednet', age_dependence: dict = None, seasonal_dependence: dict = None, insecticide: str = None, node_ids: list = None, triggered_campaign_delay: dict = None, triggers: list = None, duration: int = -1, check_eligibility_at_trigger: bool = False)
```

Add an insecticide-treated net (ITN) intervention with a seasonal usage pattern to the campaign using the **UsageDependentBednet** class. The arguments **birth_triggered** and **triggered_condition_list** are mutually exclusive. If both are provided, **triggered_condition_list** is ignored. You must add the following custom events to your config.json:

- Bednet_Discarded
- Bednet_Got_New_One
- Bednet_Using

Parameters

- **start** – The day on which to start distributing the bednets (**Start_Day** parameter).
- **coverage** – Fraction of the population receiving bed nets in a given distribution event
- **blocking_config** – The value passed gets directly assigned to the **Blocking_Config** parameter. Durations are in days. Default is **blocking_config= WaningEffectExponential(Decay_Time_Constant=730, Initial_Effect=0.9)**

This could be dictionary such as:

```
{
  "Box_Duration": 3650,
  "Initial_Effect": 0,
  "class": "WaningEffectBox"
}
```

- **killling_config** – The value passed gets directly assigned to the Killing_Config parameter. Durations are in days. Default is killing_config = WaningEffectExponential(Decay_Time_Constant=1460, Initial_Effect=0.6)

This could be dictionary such as:

```
{
  "Box_Duration": 3650,
  "Initial_Effect": 0,
  "Decay_Time_Constant": 150,
  "class": "WaningEffectBoxExponential"
}
```

- **repelling_config** – The value passed gets directly assigned to the Repelling_Config parameter. Durations are in days. Default is repelling_config = WaningEffectExponential(Decay_Time_Constant=1460, Initial_Effect=0.0)

This could be dictionary such as:

```
{
  "Box_Duration": 3650,
  "Initial_Effect": 0,
  "Decay_Time_Constant": 150,
  "class": "WaningEffectBoxExponential"
}
```

- **discard_config** – A dictionary of parameters needed to define expiration distribution. No need to define the distribution with all its parameters Default is bednet being discarded with EXPONENTIAL_DISTRIBUTION with Expiration_Period_Exponential of 10 years

Examples:

```
for Gaussian: {"Expiration_Period_Distribution": "GAUSSIAN_
↪DISTRIBUTION",
  "Expiration_Period_Gaussian_Mean": 20, "Expiration_Period_
↪Gaussian_Std_Dev":10}
for Exponential {"Expiration_Period_Distribution": "EXPONENTIAL_
↪DISTRIBUTION",
  "Expiration_Period_Exponential":150}
```

- **age_dependence** – A dictionary defining the age dependence of net use. Must contain a list of ages in years and list of usage rate. Default is uniform across all ages. Times are in years of age Examples:

```
{"Times":[], "Values":[]} or {"youth_cov":0.7, "youth_min_age":3,
↪"youth_max_age":13}
```

- **seasonal_dependence** – A dictionary defining the seasonal dependence of net use. Default is constant use during the year. Times are given in days of the year; values greater than 365 are ignored. Dictionaries can be (times, values) for linear spline or (minimum coverage, day of maximum coverage) for sinusoidal dynamics. Times are days of the year Examples:

```
{ "Times": [], "Values": [] } or { "min_cov": 0.45, "max_day": 300 }
```

- **node_ids** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of NodeSetAll.
- **birth_triggered** – If true, event is specified as a birth-triggered intervention.
- **duration** – If run as a birth-triggered event or a trigger_condition_list, specifies the duration for the distribution to continue. Default is to continue until the end of the simulation.
- **triggered_campaign_delay** – (Optional) After the trigger is received, the number of time steps until the campaign starts. Eligibility of people or nodes for the campaign is evaluated on the start day, not the triggered day. triggered_campaign_delay is a dict. Specify the actual delay distribution params, not the distribution type. E.g., { “Delay_Distribution_Constant”: 14” } Delay is in days
- **trigger_condition_list** – (Optional) A list of the events that will trigger the ITN intervention. If included, **start** is the day when monitoring for triggers begins. This argument cannot configure birth-triggered ITN (use **birth_triggered** instead).
- **ind_property_restrictions** – The IndividualProperty key:value pairs that individuals must have to receive the intervention (**Property_Restrictions_Within_Node** parameter). In the format [{ “BitingRisk”: “High” }, { “IsCool”: “Yes” }].
- **node_property_restrictions** – The NodeProperty key:value pairs that nodes must have to receive the intervention (**Node_Property_Restrictions** parameter). In the format [{ “Place”: “RURAL” }, { “ByALake”: “Yes” }]
- **check_eligibility_at_trigger** – if triggered event is delayed, you have an option to check individual/node’s eligibility at the initial trigger or when the event is actually distributed after delay.

Returns None

NOTE: Previous way of setting discard config is no longer available, you can translate it to the current way by: discard_config the old way { ‘halflife1’: 260, ‘halflife2’: 2106, ‘fraction1’: float(table_dict[‘fast_fraction’]) } discard_config translated = { “Expiration_Period_Distribution”: “DUAL_EXPONENTIAL_DISTRIBUTION”, “Expiration_Period_Mean_1”: discard_halflife, or halflife1 “Expiration_Period_Mean_2”: 365 * 40, or halflife2 “Expiration_Period_Proportion_1”: 1 or ‘fraction1’ }

#fixme This is dtk-tools commenting, needs an update for emodpy Example:

```
discard_config = { "Expiration_Period_Exponential": 10 * 365 }
age_dependence = { "Times": [0, 4, 10, 60],
                  "Values": [1, 0.9, 0.8, 0.5] }
add_ITN_age_season(config_builder, start=1, coverage=1, killing_config=killing_
    ↪ config,
                  blocking_config=blocking_config, discard_config = discard_config
                  age_dependence=age_dependence, cost=5, birth_triggered=True,
    ↪ duration=-1,
                  node_property_restrictions=[{ "Place": "Rural" }]):
```

```
emodpy_malaria.interventions.udbednet.new_intervention_as_file(camp,
                                                                start_day, file-
                                                                name=None)
```

emodpy_malaria.interventions.vaccine module

This module contains functionality for vaccine distribution.

```
emodpy_malaria.interventions.vaccine.add_scheduled_vaccine(campaign, start_day:
    int = 1, demo-
    graphic_coverage:
    float = 1.0, tar-
    get_num_individuals:
    int = None, node_ids:
    list = None, rep-
    etitions: int = 1,
    timesteps_between_repetitions:
    int = 365,
    ind_property_restrictions:
    list = None,
    node_property_restrictions:
    list = None, tar-
    get_age_min: int = 0,
    target_age_max: int
    = 125, target_gender:
    str = 'All', broad-
    cast_event: str =
    None, vaccine_type:
    str = 'Acquisition-
    Blocking', vac-
    cine_take: float = 1,
    vaccine_initial_effect:
    float = 1, vac-
    cine_box_duration:
    int = 365, vac-
    cine_exponential_decay_rate:
    float = 1.0, effi-
    cacy_is_multiplicative:
    bool = True)
```

Adds a scheduled SimpleVaccine event, with an optional BroadcastEvent, broadcast when vaccine is received.

Parameters

- **campaign** – campaign object to which the intervention will be added, and schema_path container
- **start_day** – The day the intervention is given out.
- **demographic_coverage** – This value is the probability that each individual in the target population will receive the intervention. It does not guarantee that the exact fraction of the target population set by Demographic_Coverage receives the intervention.
- **target_num_individuals** – The exact number of people to select out of the targeted group. If this value is set, demographic_coverage parameter is ignored
- **node_ids** – List of nodes to which to distribute the intervention. [] or None, indicates all nodes will get the intervention
- **repetitions** – The number of times an intervention is given, used with

timesteps_between_repetitions. -1 means the intervention repeats forever. Sets **Number_Repetitions**

- **timesteps_between_repetitions** – The interval, in timesteps, between repetitions. Ignored if repetitions = 1. Sets **Timesteps_Between_Repetitions**
- **ind_property_restrictions** – A list of dictionaries of IndividualProperties, which are needed for the individual to receive the intervention. Sets the **Property_Restrictions_Within_Node**
- **node_property_restrictions** – A list of the NodeProperty key:value pairs, as defined in the demographics file, that nodes must have to receive the intervention. Sets **Node_Property_Restrictions**
- **target_age_min** – The lower end of ages targeted for an intervention, in years. Sets **Target_Age_Min**
- **target_age_max** – The upper end of ages targeted for an intervention, in years. Sets **Target_Age_Max**
- **target_gender** – The gender targeted for an intervention: All, Male, or Female.
- **broadcast_event** – “The name of the event to be broadcast. This event must be set in the **Custom_Coordinator_Events** configuration parameter. When None or Empty, nothing is broadcast.
- **vaccine_type** – The type of vaccine to distribute in a vaccine intervention. Options are: “Generic”, “TransmissionBlocking”, “AcquisitionBlocking”, “MortalityBlocking”
- **vaccine_take** – The rate at which delivered vaccines will successfully stimulate an immune response and achieve the desired efficacy.
- **vaccine_initial_effect** – Initial efficacy of the vaccine, before decay.
- **vaccine_box_duration** – Duration in days of initial efficacy of vaccine before it starts to decay.
- **vaccine_exponential_decay_rate** – The exponential rate of vaccine efficacy after the box duration.
- **efficacy_is_multiplicative** – The overall vaccine efficacy when individuals receive more than one vaccine. When set to true (1), the vaccine efficacies are multiplied together; when set to false (0), the efficacies are additive.

Returns Nothing

```
emodpy_malaria.interventions.vaccine.new_intervention_as_file(campaign,  
                                                                start_day, file-  
                                                                name=None)
```

Write a campaign file to disk with a single bednet event, using defaults. Useful for testing and learning.

Parameters

- **campaign** – The `emod_api.campaign` object to which the intervention will be added.
- **start_day** – The day of the simulation on which the bednets are distributed. We recommend aligning this with the start of the simulation.
- **filename** – The campaign filename; can be omitted and default will be used and returned to user.

Returns The campaign filename written to disk.

emodpy_malaria.reporters package

Submodules

emodpy_malaria.reporters.builtin module

```
emodpy_malaria.reporters.builtin.add_report_vector_genetics(task, manifest,
                                                             start_day: int =
                                                             0, duration_days:
                                                             int = 365000,
                                                             nodes: list = None,
                                                             species: str = None,
                                                             gender: str = 'VEC-
                                                             TOR_FEMALE', in-
                                                             clude_vector_state:
                                                             int = 1, strat-
                                                             ify_by: str =
                                                             'GENOME', com-
                                                             bine_similar_genomes:
                                                             int = 0, spe-
                                                             cific_genome_combinations_for_stratification:
                                                             list = None, al-
                                                             lele_combinations_for_stratification:
                                                             list = None, alle-
                                                             les_for_stratification:
                                                             list = None, re-
                                                             port_description: str
                                                             = "")
```

Adds ReportVectorGenetics to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start reporting data
- **duration_days** – number of days over which to report data
- **nodes** – the list of nodes in which to collect data, empty or None means all nodes
- **species** – the species to include information on
- **gender** – gender of species to include information on. Default: “VECTOR_FEMALE”, other options: “VECTOR_MALE”, “VECTOR_BOTH_GENDERS”
- **include_vector_state** – if 1(true), adds the columns for vectors in the different states (i.e Eggs, Larva, etc)
- **stratify_by** – the way to stratify data. Default: “GENOME”, other options: “SPECIFIC_GENOME”, “ALLELE”, “ALLELE_FREQ”
- **combine_similar_genomes** – if 1(true), genomes are combined if for each locus (ignoring gender) the set of allele of the two genomes are the same (i.e. 1-0 is similar to 0-1). Depends on: “GENOME”, “SPECIFIC_GENOME” specific_genome_combinations_for_stratification: if stratifying by “SPECIFIC_GENOME”, then use these genomes to stratify by. Example:

```
{ "Allele_Combination": [[ "a0", "*" ], [ "b1", "b0" ] ] },
{ "Allele_Combination": [[ "a1", "a0" ], [ "b0", "*" ] ] }
```

- **allele_combinations_for_stratification** – if stratifying by “ALLELE”, then also add these allele name combos to the stratification, Example:

```
[[ "a0", "b0" ], [ "a1", "b1" ]]
```

- **alleles_for_stratification** – For example:

```
[ "a0", "a1", "b0", "b1" ]
```

- **report_description** – adds the description to the filename of the report to differentiate it from others

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_vector_stats(task, manifest,
species_list: list = None,
stratify_by_species: int =
0, include_wolbachia: int
= 0, include_gestation:
int = 0)
```

Adds ReportVectorStats report to the simulation. See class definition for description of the report. :param task: task to which to add the reporter, if left as None, reporter is returned (used for unittests) :param manifest: schema path file :param species_list: a list of species to include information on :param stratify_by_species: if 1(true), data will break out each the species for each node :param include_wolbachia: if 1(true), add a column for each type of Wolbachia :param include_gestation: if 1(true), adds columns for feeding and gestation

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_malaria_summary_report(task, manifest,
start_day: int = 0,
duration_days: int
= 365000, nodes:
list = None, report-
ing_interval: float
= 1, age_bins: list
= None, individ-
ual_property_filter:
str = "", infec-
tiousness_bins:
list = None,
max_number_reports:
int = 100, para-
sitemia_bins: list =
None, pretty_format:
int = 0, re-
port_description: str
= "")
```

Adds MalariaSummaryReport to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file

- **start_day** – the day of the simulation to starts collecting data for the report
- **duration_days** – The duration of simulation days over which to report events. The report will stop collecting data when the simulation day is greater than Start_Day + Duration_Days.
- **nodes** – a list of nodes from which to collect data for the report
- **reporting_interval** – Defines the cadence of the report by specifying how many time steps to collect data before writing to the file
- **age_bins** – The max age in years per bin, listed in ascending order. Use a large value for the last bin, to collect all remaining individuals
- **individual_property_filter** – the individual ‘property:value’ to filter on. The default of an empty string means the report is not filtered. Example: “Risk:HIGH”
- **infectiousness_bins** – infectiousness Bins to aggregate within for the report
- **max_number_reports** – the maximum number of report output files that will be produced for a given simulation
- **parasitemia_bins** – Parasitemia bins on which to aggregate. A value ≤ 0 in the first bin indicates that uninfected individuals are added to this bin. You must sort your input data from low to high.
- **pretty_format** – if 1(true) sets pretty JSON formatting, which includes carriage returns, line feeds, and spaces for easier readability. The default, 0 (false), saves space where everything is on one line.
- **report_description** – adds the description to the filename of the report to differentiate it from others

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_malaria_patient_json_report(task, manifest)
```

Adds MalariaPatientJSONReport report to the simulation. See class definition for description of the report. You do not need to configure any data parameters to generate this report. :param task: task to which to add the reporter, if left as None, reporter is returned (used for unittests) :param manifest: schema path file

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_malaria_transmission_report(task,
                                                                    manifest,
                                                                    start_day:
                                                                    int = 0, duration_days:
                                                                    int = 365000,
                                                                    nodes:
                                                                    list = None,
                                                                    pretty_format:
                                                                    int = 0, report_description:
                                                                    str = "")
```

Adds ReportSimpleMalariaTransmissionJSON report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)

- **manifest** – schema path file
- **start_day** – the day to start collecting data for the report.
- **duration_days** – The duration of simulation days over which to report events. The report will stop collecting data when the simulation day is greater than Start_Day + Duration_Days
- **nodes** – list of nodes for which to collect data for the report
- **pretty_format** – if 1(true) sets pretty JSON formatting, which includes carriage returns, line feeds, and spaces for easier readability. The default, 0 (false), saves space where everything is on one line.
- **report_description** – adds the description to the filename of the report to differentiate it from others

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_malaria_filtered(task, manifest,
                                                             start_day: int =
                                                             0, end_day: int
                                                             = 365000, nodes:
                                                             list = None, re-
                                                             port_filename: str
                                                             = 'ReportMalar-
                                                             iaFiltered.json',
                                                             min_age_years:
                                                             float = 0,
                                                             max_age_years:
                                                             float = 125,
                                                             has_interventions:
                                                             list = None, in-
                                                             clude_30day_avg_infection_duration:
                                                             bool = True)
```

Adds ReportMalariaFiltered report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **end_day** – the day of simulation to stop collecting data
- **nodes** – list of nodes for which to collect the data, None or [] collects all the nodes
- **report_filename** – name of the file to be written
- **min_age_years** – Minimum age in years of people to collect data on
- **max_age_years** – Maximum age in years of people to collect data on
- **has_interventions** – A channel is added to the report for each InterventionName provided. The channel name will be Has_<InterventionName> and will be the fraction of the population that has that intervention. The **Intervention_Name** in the campaign should be the values in this parameter
- **include_30day_avg_infection_duration** – If set to True (1), the 30-Day Avg Infection channel is included in the report

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_spatial_report_malaria_filtered(task,
                                                                    manifest,
                                                                    start_day:
                                                                    int = 0,
                                                                    end_day:
                                                                    int =
                                                                    365000,
                                                                    reporting_interval:
                                                                    int = 1,
                                                                    nodes:
                                                                    list =
                                                                    None,
                                                                    report_filename:
                                                                    str =
                                                                    'SpatialReport-MalariaFiltered',
                                                                    spatial_output_channels:
                                                                    list =
                                                                    None)
```

Adds SpatialReportMalariaFiltered report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **end_day** – the day of simulation to stop collecting data
- **reporting_interval** – defines the cadence of the report by specifying how many time steps to collect data before writing to the file.
- **nodes** – list of nodes for which to collect the data
- **report_filename** – name of the file to be written
- **spatial_output_channels** – list of names of channels you want to have output for. Available channels are: “Adult_Vectors”, “Air_Temperature”, “Births”, “Blood_Smear_Gametocyte_Prevalence”, “Blood_Smear_Parasite_Prevalence”, “Campaign_Cost”, “Daily_Bites_Per_Human”, “Daily_EIR”, “Disease_Deaths”, “Fever_Prevalence”, “Human_Infectious_Reservoir”, “Infected”, “Infectious_Vectors”, “Land_Temperature”, “Mean_Parasitemia”, “New_Clinical_Cases”, “New_Infections”, “New_Reported_Infections”, “New_Severe_Cases”, “PCR_Gametocyte_Prevalence”, “PCR_Parasite_Prevalence”, “PfHRP2_Prevalence”, “Population”, “Prevalence”, “Rainfall”, “Relative_Humidity”, “True_Prevalence” Defaults: [“Blood_Smear_Parasite_Prevalence”, “New_Clinical_Cases”, “Population”]

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_event_counter(task, manifest,
start_day: int = 0, duration_days: int = 365000,
event_trigger_list: list = None, nodes: list = None, re-
port_description: str = ")")
```

Adds ReportEventCounter report to the simulation. See class definition for description of the report. :param task: task to which to add the reporter, if left as None, reporter is returned (used for unittests) :param manifest: schema path file :param start_day: the day of the simulation to start counting events :param duration_days: number of days for which to count events :param event_trigger_list: list of events which to count :param nodes: list of nodes in which to count the events :param report_description: used by reports and custom reports. Augments the filename of the report.

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_malaria_sql_report(task, manifest, start_day:
int = 0, end_day: int = 365000, in-
clude_infection_table: int = 1, in-
clude_health_table: int = 1, include_drug_table: int = 0)
```

Adds MalariaSqlReport report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **end_day** – the day of the simulation to stop collecting data
- **include_infection_table** – if 1(true), include the table that provides data at each time step for each active infection
- **include_health_table** – if 1(true), include the table that provides data at each time step for a person's health
- **include_drug_status** – if 1(true), include the table that provides data at each time step for each drug used

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_vector_habitat_report(task, manifest)
```

Adds VectorHabitatReport report to the simulation. See class definition for description of the report. You do not need to configure any data parameters to generate this report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_malaria_immunity_report (task, manifest,
                                                              start_day: int = 0,
                                                              duration_days: int
                                                              = 365000, reporting_interval:
                                                              int = 1,
                                                              max_number_reports:
                                                              int = 365000,
                                                              nodes: list =
                                                              None, age_bins:
                                                              list = None,
                                                              pretty_format:
                                                              int = 0, report_description:
                                                              str = ")
Add MalariaImmunityReport report to the simulation. See class definition for description of the report.
```

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **duration_days** – number of days over which to report data
- **reporting_interval** – defines the cadence of the report by specifying how many time steps to collect data before writing to the file.
- **max_number_reports** – the maximum number of report output files that will be produced for a given simulation
- **nodes** – list of nodes for which to collect data
- **age_bins** – The max age in years per bin, listed in ascending order. Use a large value for the last bin, to collect all remaining individuals
- **pretty_format** – if 1(true) sets pretty JSON formatting, which includes carriage returns, line feeds, and spaces for easier readability. The default, 0 (false), saves space where everything is on one line.
- **report_description** – adds the description to the filename of the report to differentiate it from others

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_malaria_survey_analyzer(task, manifest,
                                                             start_day: int =
                                                             0, duration_days:
                                                             int = 365000,
                                                             event_trigger_list:
                                                             list = None, report-
                                                             ing_interval:
                                                             float = 1,
                                                             max_number_reports:
                                                             int = 365000,
                                                             nodes: list =
                                                             None, individ-
                                                             ual_property_to_collect:
                                                             str = "",
                                                             pretty_format:
                                                             int = 0, re-
                                                             port_description:
                                                             str = ")")
```

Adds MalariaSurveyJSONAnalyzer report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **duration_days** – number of days over which to report data
- **reporting_interval** – defines the cadence of the report by specifying how many time steps to collect data before writing to the file
- **event_trigger_list** – list of individual events to include into the report
- **max_number_reports** – the maximum number of report output files that will be produced for a given simulation
- **nodes** – list of nodes for which to collect data
- **individual_property_to_collect** – name of the Individual Property Key whose value to collect. Empty string means collect values for all Individual Properties
- **pretty_format** – if 1(true) sets pretty JSON formatting, which includes carriage returns, line feeds, and spaces for easier readability. The default, 0 (false), saves space where everything is on one line
- **report_description** – adds the description to the filename of the report to differentiate it from others

Returns:

```
emodpy_malaria.reporters.builtin.add_drug_status_report(task, manifest, start_day:
                                                         int = 0, end_day: int =
                                                         365000)
```

Adds ReportDrugStatus report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)

- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **end_day** – the day of the simulation to stop collecting data

Returns Nothing

`emodpy_malaria.reporters.builtin.add_human_migration_tracking(task, manifest)`

Adds ReportHumanMigrationTracking report to the simulation. There are no special parameter that need to be configured to generate the report. However, the simulation must have migration enabled.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file

Returns Nothing

`emodpy_malaria.reporters.builtin.add_report_node_demographics(task, manifest, age_bins: list = None, individual_property_to_collect: str = "", stratify_by_gender: int = 1)`

Adds ReportNodeDemographics report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **age_bins** – the age bins (in years) to aggregate within and report. An empty array does not stratify by age. You must sort your input data from low to high.
- **individual_property_to_collect** – The name of the IndividualProperties key by which to stratify the report. An empty string does not stratify by Individual Properties
- **stratify_by_gender** – if 1(true), to stratify by gender. Set to false (0) to not stratify by gender.

Returns Nothing

`emodpy_malaria.reporters.builtin.add_report_node_demographics_malaria(task, manifest, age_bins: list = None, individual_property_to_collect: str = "", stratify_by_gender: int = 1)`

Adds ReportNodeDemographicsMalaria report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **age_bins** – the age bins (in years) to aggregate within and report. An empty array does not stratify by age. You must sort your input data from low to high.
- **individual_property_to_collect** – The name of theIndividualProperties key by which to stratify the report. An empty string does not stratify by Individual Properties
- **stratify_by_gender** – if 1(true), to stratify by gender. Set to false (0) to not stratify by gender.

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_node_demographics_malaria_genetics(task,  
man-  
i-  
fest,  
bar-  
codes:  
list  
=  
None,  
drug_resistant_s  
list  
=  
None,  
drug_resistant_s  
str  
=  
'NUM_PEOPLE',  
age_bins:  
list  
=  
None,  
in-  
di-  
vid-  
ual_property_to_  
str  
=  
",  
strat-  
ify_by_gender:  
int  
=  
1)
```

Adds ReportNodeDemographicsMalariaGenetics report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **barcodes** – a list of barcode strings. The report contains the number of human infections with each barcode. Use '*' for a wild card at a loci to include all values at that loci. For example, "A*T" includes AAT, ACT, AGT, and ATT. The report contains a BarcodeOther column for barcodes that are not defined. Note: There is no validation that the barcode strings are valid barcodes for the scenario.
- **drug_resistant_strings** – a list of strings representing the set of drug resistant markers. A column will be created with the number of humans infections with that barcode. One can use '*' for a wild card. A 'BarcodeOther' column will be created for barcodes not define
- **drug_resistant_statistic_type** – indicates the statistic in the Drug Resistant columns: NUM_PEOPLE_WITH_RESISTANT_INFECTION = A person is counted if they have one infection with that drug resistant marker; NUM_INFECTIONS = The total number of infections with that marker.
- **age_bins** – the age bins (in years) to aggregate within and report. An empty array does not stratify by age. You must sort your input data from low to high.
- **individual_property_to_collect** – The name of theIndividualProperties key by which to stratify the report. An empty string does not stratify by Individual Properties
- **stratify_by_gender** – if 1(true), to stratify by gender. Set to false (0) to not stratify by gender.

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_vector_migration(task, manifest,
                                                             start_day: int =
                                                             0, end_day: int =
                                                             365000)
```

Adds ReportVectorMigration report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **start_day** – the day of the simulation to start collecting data
- **end_day** – the day of the simulation to stop collecting data

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_vector_stats_malaria_genetics(task,
                                                                           manifest,
                                                                           species_list:
                                                                           list
                                                                           =
                                                                           None,
                                                                           strat-
                                                                           ify_by_species:
                                                                           int
                                                                           =
                                                                           0,
                                                                           include_wolbachia:
                                                                           int
                                                                           =
                                                                           0,
                                                                           include_gestation:
                                                                           int
                                                                           =
                                                                           0,
                                                                           bar-
                                                                           codes:
                                                                           list
                                                                           =
                                                                           None)
```

Adds ReportVectorStatsMalariaGenetics report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – schema path file
- **species_list** – a list of species to include information on
- **stratify_by_species** – if 1(true), data will break out each the species for each node
- **include_wolbachia** – if 1(true), add a column for each type of Wolbachia
- **include_gestation** – if 1(true), adds columns for feeding and gestation
- **barcodes** – a list of barcode strings. The report contains the number of human infections with each barcode. Use ‘*’ for a wild card at a loci to include all values at that loci. For example, “A*T” includes AAT, ACT, AGT, and ATT. The report contains a BarcodeOther column for barcodes that are not defined. Note: There is no validation that the barcode strings are valid barcodes for the scenario.

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_event_recorder(task,
                                                     event_list,
                                                     only_include_events_in_list=True,
                                                     ips_to_record=None)
```

Adds ReportEventRecorder report to the simulation. See class definition for description of the report.

Parameters

- **task** – task to which to add the reporter
- **event_list** – a list of events to record or exclude, depending on value of `only_include_events_in_list`
- **only_include_events_in_list** – if True, only record events listed. if False, record ALL events EXCEPT for the ones listed
- **ips_to_record** – list of individual properties to include in report

Returns Nothing

```
emodpy_malaria.reporters.builtin.add_report_intervention_pop_avg(task,
                                                                manifest,
                                                                start_day:
                                                                int = 0, du-
                                                                ration_days:
                                                                int      =
                                                                36500000,
                                                                re-
                                                                port_description:
                                                                str      = "",
                                                                nodes: list =
                                                                None)
```

Adds ReportInterventionPopAvg reporter. See class definition for description of the report.

Parameters

- **task** – Task to which to add the reporter, if left as None, reporter is returned (used for unittests)
- **manifest** – Schema path file
- **start_day** – The day of the simulation to start collecting data
- **duration_days** – The number of days over which to collect report data
- **report_description** – Augments the filename of the report. If multiple CSV reports are being generated, this allows you to distinguish among the multiple reports
- **nodes** – List of nodes for which to collect data

Returns Nothing

```
class emodpy_malaria.reporters.builtin.ReportVectorGenetics (class_name: str =
                                                                None, parameters:
                                                                dict = <factory>,
                                                                Enabled: bool =
                                                                True, Pretty_Format:
                                                                bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The vector genetics report is a CSV-formatted report (ReportVectorGenetics.csv) that collects information on how many vectors of each genome/allele combination exist at each time, node, and vector state. Information can only be collected on one species per report.

config (`config_builder`, `manifest`)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportVectorStats (class_name: str = None,  
                                                    parameters: dict = <factory>,  
                                                    Enabled: bool  
                                                    = True, Pretty_Format:  
                                                    bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The vector statistics report is a CSV-formatted report (`ReportVectorStats.csv`) that provides detailed life-cycle data on the vectors in the simulation. The report is stratified by time, node ID, and (optionally) species.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.MalariaSummaryReport (class_name: str =  
                                                    None, parameters:  
                                                    dict = <factory>,  
                                                    Enabled: bool =  
                                                    True, Pretty_Format:  
                                                    bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The population-level malaria summary report is a JSON-formatted report (`MalariaSummaryReport.json`) that provides a summary of malaria data across the population. The data are grouped into different bins such as age, parasitemia, and infectiousness.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.MalariaPatientJSONReport (class_name:  
                                                    str = None, pa-  
                                                    rameters: dict  
                                                    = <factory>,  
                                                    Enabled:  
                                                    bool = True,  
                                                    Pretty_Format:  
                                                    bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The malaria patient data report is a JSON-formatted report (`MalariaPatientReport.json`) that provides medical data for each individual on each day of the simulation. For example, for a specified number of time steps, each “patient” has information collected on the temperature of their fever, their parasite counts, treatments they received, and other relevant data.

config (*config_builder, manifest*)

parameters: `dict`

```

class emodpy_malaria.reporters.builtin.ReportSimpleMalariaTransmissionJSON (class_name:
    str
    =
    None,
    pa-
    ram-
    e-
    ters:
    dict
    =
    <fac-
    tory>,
    En-
    abled:
    bool
    =
    True,
    Pretty_Format:
    bool
    =
    True)

```

Bases: `emodpy.reporters.base.BuiltInReporter`

The simple malaria transmission report (`ReportSimpleMalariaTransmissionJSON.json`) is a JSON-formatted report that provides data on malaria transmission, by tracking who transmitted malaria to whom. The report can only be used when the simulation setup parameter **Malaria_Model** is set to `MALARIA_MECHANISTIC_MODEL_WITH_CO_TRANSMISSION`. This report is typically used as input to the GenEpi model.

config (*config_builder, manifest*)

parameters: `dict`

```

class emodpy_malaria.reporters.builtin.ReportMalariaFiltered (class_name:  str
    = None, param-
    eters:  dict  =
    <factory>,  En-
    abled: bool = True,
    Pretty_Format:
    bool = True)

```

Bases: `emodpy.reporters.base.BuiltInReporter`

The malaria filtered report (`ReportMalariaFiltered.json`) is the same as the default InsetChart report, but provides filtering options to enable the user to select the data to be displayed for each time step or for each node. See InsetChart for more information about InsetChart.json.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.SpatialReportMalariaFiltered(class_name:
                                                                    str      =
                                                                    None,
                                                                    parameters: dict
                                                                    = <factory>,
                                                                    Enabled:
                                                                    bool
                                                                    = True,
                                                                    Pretty_Format:
                                                                    bool      =
                                                                    True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The filtered malaria spatial report (`SpatialReportMalariaFiltered.bin`) provides spatial information on malaria simulations and allows for filtering the data and collection over different intervals. This report is similar to the Spatial output report but allows for data collection and filtering over different intervals using the `Start_Day` and a `Reporting_Interval` parameters

config (*config_builder*, *manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportEventCounter(class_name:  str =
                                                            None,      parameters:
                                                            dict = <factory>,
                                                            Enabled:  bool = True,
                                                            Pretty_Format: bool =
                                                            True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The event counter report is a JSON-formatted file (`ReportEventCounter.json`) that keeps track of how many of each event types occurs during a time step. The report produced is similar to the `InsetChart.json` channel report, where there is one channel for each event defined in the configuration file (`config.json`).

config (*config_builder*, *manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.MalariaSqlReport(class_name: str = None,
                                                         parameters: dict = <factory>,
                                                         Enabled:  bool =
                                                         True, Pretty_Format: bool
                                                         = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The MalariaSQL report outputs epidemiological and transmission data. Because of the quantity and complexity of the data, the report output is a multi-table SQLite relational database (see <https://sqlitebrowser.org/>). Use the configuration parameters to manage the size of the database.

config (*config_builder*, *manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.VectorHabitatReport(class_name:  str =
                                                            None,      parameters:
                                                            dict = <factory>,
                                                            Enabled:  bool = True,
                                                            Pretty_Format: bool
                                                            = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The vector habitat report is a JSON-formatted file (`VectorHabitatReport.json`) containing habitat data for each vector species included in the simulation. It focuses on statistics relevant to mosquito developmental stages (e.g. eggs and larvae), such as egg capacity and larval crowding.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.MalariaImmunityReport (class_name: str
                                                             = None, parameters: dict =
                                                             <factory>, Enabled: bool = True,
                                                             Pretty_Format: bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The malaria immunity report is a JSON-formatted file (`MalariaImmunityReport.json`) that provides statistics for several antibody types for specified age bins over a specified reporting duration. Specifically, the report tracks the average and standard deviation in the fraction of observed antibodies for merozoite surface protein (MSP), *Plasmodium falciparum* erythrocyte membrane protein 1 (PfEMP1), and non-specific (and less immunogenic) minor surface epitopes. The total possible is determined by parameters `Falciparum_MSP_Variants`, `Falciparum_PfEMP1_Variants`, and `Falciparum_Nonspecific_Types`. The greater the fraction, the more antibodies the individual has against possible new infections. The smaller the fraction, the more naïve the individual's immune system is to malaria.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.MalariaSurveyJSONAnalyzer (class_name: str = None,
                                                                    parameters: dict =
                                                                    <factory>,
                                                                    Enabled: bool = True,
                                                                    Pretty_Format: bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportDrugStatus (class_name: str = None,
                                                           parameters: dict = <factory>,
                                                           Enabled: bool = True,
                                                           Pretty_Format: bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The drug status report provides status information on the drugs that an individual has taken or is waiting to take. Because the report provides information for each drug, for each individual, and for each time step, you may want to use the `Start_Day` and `End_Day` parameters to limit the size the output file.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportHumanMigrationTracking (class_name:
                                                                    str      =
                                                                    None,
                                                                    param-
                                                                    eters: dict
                                                                    = <fac-
                                                                    tory>,
                                                                    Enabled:
                                                                    bool
                                                                    = True,
                                                                    Pretty_Format:
                                                                    bool    =
                                                                    True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The human migration tracking report is a CSV-formatted report (ReportHumanMigrationTracking.csv) that provides details about human travel during simulations. The report provides one line for each surviving individual who migrates during the simulation.

config (*config_builder*, *manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportNodeDemographics (class_name:
                                                                    str      = None,
                                                                    parameters:
                                                                    dict    = <fac-
                                                                    tory>,
                                                                    Enabled:
                                                                    bool    = True,
                                                                    Pretty_Format:
                                                                    bool    = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

The node demographics report is a CSV-formatted report (ReportNodeDemographics.csv) that provides population information stratified by node. For each time step, the report collects data on each node and age bin.

config (*config_builder*, *manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportNodeDemographicsMalaria (class_name:
                                                                    str      =
                                                                    None,
                                                                    param-
                                                                    eters:
                                                                    dict    =
                                                                    <fac-
                                                                    tory>,
                                                                    En-
                                                                    abled:
                                                                    bool    =
                                                                    True,
                                                                    Pretty_Format:
                                                                    bool    =
                                                                    True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

This report extends the data collected in the ReportNodeDemographics by adding data about the number of

infections with specific barcodes. The malaria node demographics genetics report does not include columns for Genome_Markers because this report assumes that the simulation setup parameter Malaria_Model is set to MALARIA_MECHANISTIC_MODEL_WITH_PARASITE_GENETICS.

Note: If you need detailed data on the infections with different barcodes, use the MalariaSqlReport. That report contains data on all barcodes, without specifying what they are.

config (*config_builder, manifest*)

parameters: *dict*

```
class emodpy_malaria.reporters.builtin.ReportNodeDemographicsMalariaGenetics (class_name:
                                                                    str
                                                                    =
                                                                    None,
                                                                    pa-
                                                                    ram-
                                                                    e-
                                                                    ters:
                                                                    dict
                                                                    =
                                                                    <fac-
                                                                    tory>,
                                                                    En-
                                                                    abled:
                                                                    bool
                                                                    =
                                                                    True,
                                                                    Pretty_Format:
                                                                    bool
                                                                    =
                                                                    True)
```

Bases: *emodpy.reporters.base.BuiltInReporter*

This report extends the data collected in the ReportNodeDemographics by adding data about the number of infections with specific barcodes. The malaria node demographics genetics report does not include columns for Genome_Markers because this report assumes that the simulation setup parameter Malaria_Model is set to MALARIA_MECHANISTIC_MODEL_WITH_PARASITE_GENETICS.

Note: If you need detailed data on the infections with different barcodes, use the MalariaSqlReport. That report contains data on all barcodes, without specifying what they are.

config (*config_builder, manifest*)

parameters: *dict*

```
class emodpy_malaria.reporters.builtin.ReportVectorMigration (class_name:  str
                                                                    = None, param-
                                                                    eters:  dict  =
                                                                    <factory>,  En-
                                                                    abled: bool = True,
                                                                    Pretty_Format:
                                                                    bool = True)
```

Bases: *emodpy.reporters.base.BuiltInReporter*

This report provides detailed information on where and when vectors are migrating. Because there can be one line for each migrating vector, you may want to use the Start_Day and End_Day parameters to limit the size the output file.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportVectorStatsMalariaGenetics (class_name:
    str
    =
    None,
    pa-
    ram-
    e-
    ters:
    dict
    =
    <fac-
    tory>,
    En-
    abled:
    bool
    =
    True,
    Pretty_Format:
    bool
    =
    True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

This report extends the data collected in the ReportVectorStats by adding data about the number of infections with specific barcodes. The malaria node demographics genetics report does not include columns for Genome_Markers because this report assumes that the simulation setup parameter Malaria_Model is set to MALARIA_MECHANISTIC_MODEL_WITH_PARASITE_GENETICS.

config (*config_builder, manifest*)

parameters: `dict`

```
class emodpy_malaria.reporters.builtin.ReportInterventionPopAvg (class_name:
    str = None, pa-
    rameters: dict
    = <factory>,
    Enabled:
    bool = True,
    Pretty_Format:
    bool = True)
```

Bases: `emodpy.reporters.base.BuiltInReporter`

ReportInterventionPopAvg is a CSV-formatted report that gives population average data on the usage of interventions. It provides data on the fraction of people or nodes that have an intervention as well as averages on the intervention's efficacy. For each persistent intervention that has been distributed to a node or person, the report provides one line in the CSV for each intervention used in that node. Since node-level intervention (usually vector control) can only have one per node, the data will be for that one intervention. The individual-level interventions will have data for the people in that node.

config (*config_builder, manifest*)

parameters: `dict`

3.1.2 Submodules

emodpy_malaria.malaria_config module

`emodpy_malaria.malaria_config.get_file_from_http(url)`

Get data files from simple http server.

`emodpy_malaria.malaria_config.set_team_defaults(config, manifest)`

Set configuration defaults using team-wide values, including drugs and vector species.

`emodpy_malaria.malaria_config.set_team_drug_params(config, manifest)`

`emodpy_malaria.malaria_config.set_parasite_genetics_params(config, manifest, var_gene_randomness_type: str = 'ALL_RANDOM')`

Sets up the default parameters for parasite genetics simulations Malaria_Model = "MALARIA_MECHANISTIC_MODEL_WITH_PARASITE_GENETICS"

Parameters

- **config** –
- **manifest** – schema path container
- **var_gene_randomness_type** – possible values are "FIXED_NEIGHBORHOOD", "FIXED_MSP", "ALL_RANDOM" (default)

Returns configured config

`emodpy_malaria.malaria_config.get_drug_params(cb, drug_name)`

`emodpy_malaria.malaria_config.set_drug_param(config, drug_name: str = None, parameter: str = None, value: any = None)`

Set a drug parameter, by passing in drug name, parameter and the parameter value. Added to facilitate adding drug Resistances, **Example:**

```
artemether_drug_resistance = [{
    "Drug_Resistant_String": "A",
    "PKPD_C50_Modifier": 2.0,
    "Max_IRBC_Kill_Modifier": 0.9}]
set_drug_param(cb, drug_name='Artemether', parameter="Resistances",
    value=artemether_drug_resistance)
```

Parameters

- **config** –
- **drug_name** – The drug that has a parameter to set
- **parameter** – The parameter to set
- **value** – The new value to set

Returns Nothing or error if drug name is not found

`emodpy_malaria.malaria_config.add_drug_resistance(config, manifest, drugname: str = None, drug_resistant_string: str = None, pkpd_c50_modifier: float = 1.0, max_irbc_kill_modifier: float = 1.0)`

Adds drug resistances by drug name and parameters

Parameters

- **config** –
- **manifest** –
- **drugname** – name of the drug for which to assign resistances
- **drug_resistant_string** – A series of nucleotide base letters (A, C, G, T) that represent the drug resistant values at locations in the genome
- **pkpd_c50_modifier** – If the parasite has this genome marker, this value will be multiplied times the ‘Drug_PKPD_C50’ value of the drug. Genomes with multiple markers will be simply multiplied together
- **max_irbc_kill_modifier** – If the parasite has this genome marker, this value will be multiplied times the ‘Max_Drug_IRBC_Kill’ value of the drug. Genomes with multiple markers will be simply multiplied together

Returns configured config

```
emodpy_malaria.malaria_config.set_species_param(config, species, parameter, value, overwrite=False)
```

Pass through for vector version of function.

```
emodpy_malaria.malaria_config.add_species(config, manifest, species_to_select)
```

Pass through for vector version of function.

```
emodpy_malaria.malaria_config.add_insecticide_resistance(config, manifest, insecticide_name: str = "", species: str = "", allele_combo: list = None, blocking: float = 1.0, killing: float = 1.0, repelling: float = 1.0, larval_killing: float = 1.0)
```

Pass through for vector version of function.

```
emodpy_malaria.malaria_config.get_species_params(config, species: str = None)
```

Pass through for vector version of function.

```
emodpy_malaria.malaria_config.set_max_larval_capacity(config, species_name: str, habitat_type: str, max_larval_capacity: int)
```

Set the Max_Larval_Capacity for a given species and habitat. Effectively doing something like: `simulation.task.config.parameters.Vector_Species_Params[i][“Habitats”][j][“Max_Larval_Capacity”] = max_larval_capacity` where `i` is index of `species_name` and `j` is index of `habitat_type`.

Parameters

- **config** –
- **species_name** – string. Species_Name to target.
- **habitat_type** – enum. Habitat_Type to target.
- **max_larval_capacity** – integer. New value of Max_Larval_Capacity.

Returns Nothing.

emodpy_malaria.malaria_vector_species_params module

emodpy_malaria.malaria_vector_species_params.**species_params** (*manifest, species*)

Returns configured species parameters based on species name

Parameters

- **manifest** – file that contains path to the schema file
- **species** – species, configuration for which, we will be adding to the simulation.

Returns Configured species parameters

emodpy_malaria.vector_config module

emodpy_malaria.vector_config.**set_team_defaults** (*config, manifest*)

Set configuration defaults using team-wide values, including drugs and vector species.

Parameters

- **config** –
- **manifest** –

Returns configured config

emodpy_malaria.vector_config.**get_species_params** (*config, species: str = None*)

Returns the species parameters dictionary with the matching species **Name**

Parameters

- **config** –
- **species** – Species to look up

Returns Dictionary of species parameters with the matching name

emodpy_malaria.vector_config.**set_species_param** (*config, species, parameter, value, overwrite=False*)

Sets a parameter value for a specific species. Raises value error if species not found

Parameters

- **config** –
- **species** – name of species for which to set the parameter
- **parameter** – parameter to set
- **value** – value to set the parameter to
- **overwrite** – if set to True and parameter is a list, overwrites the parameter with value, appends by default

Returns Nothing

emodpy_malaria.vector_config.**add_species** (*config, manifest, species_to_select*)

Adds species with preset parameters from ‘malaria_vector_species_params.py’, if species name not found - “gambiae” parameters are added and the new species name assigned.

Parameters

- **config** – schema-backed config smart dict

- **manifest** –
- **species_to_select** – a list of species or a name of a single species you'd like to set from `malaria_vector_species_params.py`

Returns configured config

`emodpy_malaria.vector_config.add_genes_and_alleles` (*config*, *manifest*, *species*: *str* = *None*, *alleles*: *list* = *None*)

Adds alleles to a species

Example:

```
"Genes": [
  {
    "Alleles": [
      {
        "Name": "X1",
        "Initial_Allele_Frequency": 0.5,
        "Is_Y_Chromosome": 0
      },
      {
        "Name": "X2",
        "Initial_Allele_Frequency": 0.25,
        "Is_Y_Chromosome": 0
      },
      {
        "Name": "Y1",
        "Initial_Allele_Frequency": 0.15,
        "Is_Y_Chromosome": 1
      },
      {
        "Name": "Y2",
        "Initial_Allele_Frequency": 0.1,
        "Is_Y_Chromosome": 1
      }
    ],
    "Is_Gender_Gene": 1,
    "Mutations": []
  }
]
```

Parameters

- **config** –
- **manifest** –
- **species** – species to which to assign the alleles
- **alleles** – List of tuples of (**Name**, **Initial_Allele_Frequency**, **Is_Y_Chromosome**) for a set of alleles or (**Name**, **Initial_Allele_Frequency**), 1/0 or True/False can be used for **Is_Y_Chromosome**, third parameter is assumed False (0). If the third parameter is set to 1 in any of the tuples, we assume, this is a gender gene. **Example:**

```
[("X1", 0.25), ("X2", 0.35), ("Y1", 0.15), ("Y2", 0.25)]
[("X1", 0.25, 0), ("X2", 0.35, 0), ("Y1", 0.15, 1), ("Y2", 0.25, 1)]
```

Returns configured config

`emodpy_malaria.vector_config.add_mutation`(*config*, *manifest*, *species*, *mutate_from*, *mutate_to*, *probability*)

Adds to **Mutations** parameter in a Gene which has the matching **Alleles**

Parameters

- **config** –
- **manifest** –
- **species** – Name of vector species to which we're adding mutations
- **mutate_from** – The allele in the gamete that could mutate
- **mutate_to** – The allele that this locus will change to during gamete generation
- **probability** – The probability that the allele will mutate from one allele to the other during the creation of the gametes

Returns configured config

`emodpy_malaria.vector_config.add_trait`(*config*, *manifest*, *species*, *allele_combo*: *list* = *None*, *trait_modifiers*: *list* = *None*)

Use this function to add traits as part of vector genetics configuration, the trait is assigned to the species' **Gene_To_Trait_Modifiers** parameter Should produce something like **Example**:

```
{
  "Allele_Combinations" : [
    [ "X", "X" ],
    [ "a0", "a1" ]
  ],
  "Trait_Modifiers" : [
    {
      "Trait" : "FECUNDITY",
      "Modifier": 0.7
    }
  ]
}
```

Parameters

- **config** –
- **manifest** –
- **species** – Name of species for which to add this **Gene_To_Trait_Modifiers**
- **allele_combo** – List of lists, This defines a possible subset of allele pairs that a vector could have. Each pair are alleles from one gene. If the vector has this subset, then the associated traits will be adjusted. Order does not matter. '*' is allowed when only the occurrence of one allele is important. **Example**:

```
[[ "X", "X" ], [ "a0", "a1" ]]
```

- **trait_modifiers** – List of tuples of (**Trait**, **Modifier**) for the *Allele_Combinations** **Example**:

```
[("FECUNDITY", 0.5), ("X_SHRED", 0.80)]
```

Returns configured config

```
emodpy_malaria.vector_config.add_insecticide_resistance(config, manifest, insecticide_name: str = "",  
                                                         species: str = "", allele_combo: list = None,  
                                                         blocking: float = 1.0,  
                                                         killing: float = 1.0, repelling: float = 1.0,  
                                                         larval_killing: float = 1.0)
```

Use this function to add to the list of **Resistances** parameter for a specific insecticide Add each resistance separately. **Example:**

```
Insecticides = [  
  {  
    "Name": "pyrethroid",  
    "Resistances": [  
      {  
        "Allele_Combinations": [  
          "a1",  
          "a1"  
        ]  
      },  
      "Blocking_Modifier": 1.0,  
      "Killing_Modifier": 0.85,  
      "Repelling_Modifier": 0.72,  
      "Larval_Killing_Modifier": 0,  
      "Species": "gambiae"  
    ]  
  },  
  { .. }  
]
```

Parameters

- **config** –
- **manifest** –
- **insecticide_name** – The name of the insecticide to which attach the resistance.
- **species** – Name of the species of vectors.
- **allele_combo** – List of combination of alleles that vectors must have in order to be resistant.
- **blocking** – The value used to modify (multiply) the blocking effectivity of an intervention.
- **killing** – The value used to modify (multiply) the killing effectivity of an intervention.
- **repelling** – The value used to modify (multiply) the repelling effectivity of an intervention.
- **larval_killing** – The value used to modify (multiply) the larval killing effectivity of an intervention.

Returns configured config


```
emodpy_malaria.vector_config.add_species_drivers (config, manifest, species: str =
None, driving_allele: str = None,
driver_type: str = 'CLASSIC',
to_copy: str = None, to_replace:
str = None, likelihood_list: list =
None, shredding_allele_required:
str = None, allele_to_shred: str
= None, allele_to_shred_to: str =
None, allele_shredding_fraction:
float = None, al-
lele_to_shred_to_surviving_fraction:
float = None)
```

Add a gene drive that propagates a particular set of alleles. Adds one **Alleles_Driven** item to the **Alleles_Driven** list, using 'driving_allele' as key if matching one already exists.

Example:

```
{
  "Driver_Type": "INTEGRAL_AUTONOMOUS",
  "Driving_Allele": "Ad",
  "Alleles_Driven": [
    {
      "Allele_To_Copy": "Ad",
      "Allele_To_Replace": "Aw",
      "Copy_To_Likelihood": [
        {
          "Copy_To_Allele": "Aw",
          "Likelihood": 0.1
        },
        {
          "Copy_To_Allele": "Ad",
          "Likelihood": 0.3
        },
        {
          "Copy_To_Allele": "Am",
          "Likelihood": 0.6
        }
      ]
    }
  ],
  {
    "Driver_Type" : "X_SHRED",
    "Driving_Allele" : "Ad",
    "Driving_Allele_Params" : {
      "Allele_To_Copy" : "Ad",
      "Allele_To_Replace" : "Aw",
      "Copy_To_Likelihood" : [
        {
          "Copy_To_Allele" : "Ad",
          "Likelihood" : 1.0
        },
        {
          "Copy_To_Allele" : "Aw",
          "Likelihood" : 0.0
        }
      ]
    }
  ],
  "Shredding_Alleles" : {
```

(continues on next page)

(continued from previous page)

```

        "Allele_Required"      : "Yw",
        "Allele_To_Shred"      : "Xw",
        "Allele_To_Shred_To"   : "Xm",
        "Allele_Shredding_Fraction": 0.97,
        "Allele_To_Shred_To_Surviving_Fraction" : 0.05
    }
}

```

Parameters

- **config** –
- **manifest** –
- **species** – Name of the species for which we're setting the drivers
- **driving_allele** – This is the allele that is known as the driver
- **driver_type** – This indicates the type of driver. CLASSIC - The driver can only drive if the one gamete has the driving allele and the other has a specific allele to be replaced INTEGRAL_AUTONOMOUS - At least one of the gametes must have the driver. Alleles can still be driven if the driving allele is in both gametes or even if the driving allele cannot replace the allele in the other gamete X_SHRED, Y_SHRED - cannot be used in the same species during one simulation/realization. The driving_allele must exist at least once in the genome for shredding to occur. If there is only one, it can exist in either half of the genome. DAISY_CHAIN - can be used for drives that do not drive themselves but can be driven by another allele.
- **to_copy** – The main allele to be copied **Allele_To_Copy**
- **to_replace** – The allele that must exist and will be replaced by the copy **Allele_To_Replace**
- **likelihood_list** – A list of tuples in format: [(Copy_To_Allele, Likelihood),(),()] to assign to **Copy_To_Likelihood** list
- **shredding_allele_required** – The genome must have this gender allele in order for shredding to occur. If the driver is X_SHRED, then the allele must be designated as a Y chromosome. If the driver is Y_SHRED, then the allele must NOT be designated as a Y chromosome
- **allele_to_shred** – The genome must have this gender allele in order for shredding to occur. If the driver is X_SHRED, then the allele must NOT be designated as a Y chromosome. If the driver is Y_SHRED, then the allele must be designated as a Y chromosome
- **allele_to_shred_to** – This is a gender allele that the 'shredding' will change the allele_to_shred into. It can be a temporary allele that never exists in the output or could be something that appears due to resistance/failures
- **allele_shredding_fraction** – This is the fraction of the alleles_to_Shred that will be converted to allele_to_shred_to. Values 0 to 1. If this value is less than 1, then some of the allele_to_shred will remain and be part of the gametes.
- **allele_to_shred_to_surviving_fraction** – A trait modifier will automatically generated for [Allele_To_Shred_To, *], the trait ADJUST_FERTILE_EGGS, and this value as its modifier. Values 0 to 1. A value of 0 implies perfect shredding such that no allele_to_Shred_To survive in the eggs. A value of 1 means all of the 'shredded' alleles survive.

Returns configured config

```
emodpy_malaria.vector_config.set_max_larval_capacity(config, species_name,  
                                                    habitat_type,  
                                                    max_larval_capacity)
```

Set the Max_Larval_Capacity for a given species and habitat. Effectively doing something like: `simulation.task.config.parameters.Vector_Species_Params[i][“Habitats”][j][“Max_Larval_Capacity”] = max_larval_capacity` where `i` is index of `species_name` and `j` is index of `habitat_type`.

Parameters

- **config** –
- **species_name** – string. Species_Name to target.
- **habitat_type** – enum. Habitat_Type to target.
- **max_larval_capacity** – integer. New value of Max_Larval_Capacity.

Returns Nothing.

PYTHON MODULE INDEX

e emodpy_malaria.reporters.builtin, 59
emodpy_malaria, 13 emodpy_malaria.vector_config, 81
emodpy_malaria.demographics, 13
emodpy_malaria.demographics.MalariaDemographics,
13
emodpy_malaria.interventions, 15
emodpy_malaria.interventions.bednet, 15
emodpy_malaria.interventions.common, 20
emodpy_malaria.interventions.community_health_worker,
23
emodpy_malaria.interventions.diag_survey,
25
emodpy_malaria.interventions.drug, 27
emodpy_malaria.interventions.drug_campaign,
28
emodpy_malaria.interventions.inpoteir,
35
emodpy_malaria.interventions.irs, 37
emodpy_malaria.interventions.ivermectin,
38
emodpy_malaria.interventions.mosquitorelease,
39
emodpy_malaria.interventions.outbreak,
40
emodpy_malaria.interventions.outdoorrestkill,
48
emodpy_malaria.interventions.scale_larval_habitats,
49
emodpy_malaria.interventions.spacespraying,
51
emodpy_malaria.interventions.sugartrap,
52
emodpy_malaria.interventions.treatment_seeking,
53
emodpy_malaria.interventions.udbednet,
54
emodpy_malaria.interventions.vaccine,
57
emodpy_malaria.malaria_config, 79
emodpy_malaria.malaria_vector_species_params,
81
emodpy_malaria.reporters, 59

A

- `add_campaign_event()` (in module *emodpy_malaria.interventions.outbreak*), 46
- `add_community_health_worker()` (in module *emodpy_malaria.interventions.community_health_worker*), 23
- `add_diagnostic_survey()` (in module *emodpy_malaria.interventions.diag_survey*), 25
- `add_drug_campaign()` (in module *emodpy_malaria.interventions.drug_campaign*), 28
- `add_drug_resistance()` (in module *emodpy_malaria.malaria_config*), 79
- `add_drug_status_report()` (in module *emodpy_malaria.reporters.builtin*), 66
- `add_event_recorder()` (in module *emodpy_malaria.reporters.builtin*), 70
- `add_fMDA()` (in module *emodpy_malaria.interventions.drug_campaign*), 33
- `add_genes_and_alleles()` (in module *emodpy_malaria.vector_config*), 82
- `add_habitat_reduction_event()` (in module *emodpy_malaria.interventions.scale_larval_habitats*), 50
- `add_human_migration_tracking()` (in module *emodpy_malaria.reporters.builtin*), 67
- `add_InputEIR()` (in module *emodpy_malaria.interventions.inputeir*), 36
- `add_insecticide_resistance()` (in module *emodpy_malaria.malaria_config*), 80
- `add_insecticide_resistance()` (in module *emodpy_malaria.vector_config*), 83
- `add_irs_housing_modification()` (in module *emodpy_malaria.interventions.irs*), 37
- `add_ITN()` (in module *emodpy_malaria.interventions.bednet*), 19
- `add_ITN_scheduled()` (in module *emodpy_malaria.interventions.bednet*), 18
- `add_ITN_triggered()` (in module *emodpy_malaria.interventions.bednet*), 19
- `add_larval_habitat_multiplier()` (in module *emodpy_malaria.demographics.MalariaDemographics.MalariaDemographics*), 14
- `add_malaria_immunity_report()` (in module *emodpy_malaria.reporters.builtin*), 65
- `add_malaria_patient_json_report()` (in module *emodpy_malaria.reporters.builtin*), 61
- `add_malaria_sql_report()` (in module *emodpy_malaria.reporters.builtin*), 64
- `add_malaria_summary_report()` (in module *emodpy_malaria.reporters.builtin*), 60
- `add_malaria_survey_analyzer()` (in module *emodpy_malaria.reporters.builtin*), 65
- `add_malaria_transmission_report()` (in module *emodpy_malaria.reporters.builtin*), 61
- `add_MDA()` (in module *emodpy_malaria.interventions.drug_campaign*), 31
- `add_MSAT()` (in module *emodpy_malaria.interventions.drug_campaign*), 32
- `add_mutation()` (in module *emodpy_malaria.vector_config*), 83
- `add_outbreak_individual()` (in module *emodpy_malaria.interventions.outbreak*), 40
- `add_outbreak_malaria_genetics()` (in module *emodpy_malaria.interventions.outbreak*), 41
- `add_outbreak_malaria_var_genes()` (in module *emodpy_malaria.interventions.outbreak*), 44
- `add_OutdoorRestKill()` (in module *emodpy_malaria.interventions.outdoorrestkill*), 48
- `add_report_event_counter()` (in module *emodpy_malaria.reporters.builtin*), 64
- `add_report_intervention_pop_avg()` (in module *emodpy_malaria.reporters.builtin*), 71
- `add_report_malaria_filtered()` (in module

emodpy_malaria.reporters.builtin), 62
 add_report_node_demographics() (in module
emodpy_malaria.reporters.builtin), 67
 add_report_node_demographics_malaria()
 (in module *emodpy_malaria.reporters.builtin*),
 67
 add_report_node_demographics_malaria_genetics()
 (in module *emodpy_malaria.reporters.builtin*),
 68
 add_report_vector_genetics() (in module
emodpy_malaria.reporters.builtin), 59
 add_report_vector_migration() (in module
emodpy_malaria.reporters.builtin), 69
 add_report_vector_stats() (in module
emodpy_malaria.reporters.builtin), 60
 add_report_vector_stats_malaria_genetics()
 (in module *emodpy_malaria.reporters.builtin*),
 69
 add_rfMDA() (in module
emodpy_malaria.interventions.drug_campaign),
 34
 add_rfMSAT() (in module
emodpy_malaria.interventions.drug_campaign),
 33
 add_scale_larval_habitats() (in module
emodpy_malaria.interventions.scale_larval_habitats),
 49
 add_scheduled_vaccine() (in module
emodpy_malaria.interventions.vaccine),
 57
 add_spatial_report_malaria_filtered()
 (in module *emodpy_malaria.reporters.builtin*),
 63
 add_species() (in module
emodpy_malaria.malaria_config), 80
 add_species() (in module
emodpy_malaria.vector_config), 81
 add_species_drivers() (in module
emodpy_malaria.vector_config), 84
 add_trait() (in module
emodpy_malaria.vector_config), 83
 add_treatment_seeking() (in module
emodpy_malaria.interventions.treatment_seeking),
 53
 add_vector_habitat_report() (in module
emodpy_malaria.reporters.builtin), 64
 AntimalarialDrug() (in module
emodpy_malaria.interventions.common),
 21
 AntimalarialDrug() (in module
emodpy_malaria.interventions.drug), 27

B

BasicBednet() (in module

emodpy_malaria.interventions.bednet), 17
 Bednet() (in module
emodpy_malaria.interventions.bednet), 16
 BednetIntervention() (in module
emodpy_malaria.interventions.bednet), 15

G

config() (*emodpy_malaria.reporters.builtin.MalariaImmunityReport*
 method), 75
 config() (*emodpy_malaria.reporters.builtin.MalariaPatientJSONReport*
 method), 72
 config() (*emodpy_malaria.reporters.builtin.MalariaSqlReport*
 method), 74
 config() (*emodpy_malaria.reporters.builtin.MalariaSummaryReport*
 method), 72
 config() (*emodpy_malaria.reporters.builtin.MalariaSurveyJSONAnalyzer*
 method), 75
 config() (*emodpy_malaria.reporters.builtin.ReportDrugStatus*
 method), 75
 config() (*emodpy_malaria.reporters.builtin.ReportEventCounter*
 method), 74
 config() (*emodpy_malaria.reporters.builtin.ReportHumanMigrationTra*
 method), 76
 config() (*emodpy_malaria.reporters.builtin.ReportInterventionPopAvg*
 method), 78
 config() (*emodpy_malaria.reporters.builtin.ReportMalariaFiltered*
 method), 73
 config() (*emodpy_malaria.reporters.builtin.ReportNodeDemographics*
 method), 76
 config() (*emodpy_malaria.reporters.builtin.ReportNodeDemographicsM*
 method), 77
 config() (*emodpy_malaria.reporters.builtin.ReportNodeDemographicsM*
 method), 77
 config() (*emodpy_malaria.reporters.builtin.ReportSimpleMalariaTrans*
 method), 73
 config() (*emodpy_malaria.reporters.builtin.ReportVectorGenetics*
 method), 71
 config() (*emodpy_malaria.reporters.builtin.ReportVectorMigration*
 method), 77
 config() (*emodpy_malaria.reporters.builtin.ReportVectorStats*
 method), 72
 config() (*emodpy_malaria.reporters.builtin.ReportVectorStatsMalariaG*
 method), 78
 config() (*emodpy_malaria.reporters.builtin.SpatialReportMalariaFilter*
 method), 74
 config() (*emodpy_malaria.reporters.builtin.VectorHabitatReport*
 method), 75

D

drug_configs_from_code() (in module
emodpy_malaria.interventions.drug_campaign),
 28

E

emodpy_malaria

module, 13

emodpy_malaria.demographics

module, 13

emodpy_malaria.demographics.MalariaDemographics

module, 13

emodpy_malaria.interventions

module, 15

emodpy_malaria.interventions.bednet

module, 15

emodpy_malaria.interventions.common

module, 20

emodpy_malaria.interventions.community_health_worker

module, 23

emodpy_malaria.interventions.diag_survey

module, 25

emodpy_malaria.interventions.drug

module, 27

emodpy_malaria.interventions.drug_campaign

module, 28

emodpy_malaria.interventions.inpoteir

module, 35

emodpy_malaria.interventions.irs

module, 37

emodpy_malaria.interventions.ivermectin

module, 38

emodpy_malaria.interventions.mosquitorelease

module, 39

emodpy_malaria.interventions.outbreak

module, 40

emodpy_malaria.interventions.outdoorrestkill

module, 48

emodpy_malaria.interventions.scale_larval_ivermectin

module, 49

emodpy_malaria.interventions.spacespraying

module, 51

emodpy_malaria.interventions.sugartrap

module, 52

emodpy_malaria.interventions.treatment_seeking

module, 53

emodpy_malaria.interventions.udbednet

module, 54

emodpy_malaria.interventions.vaccine

module, 57

emodpy_malaria.malaria_config

module, 79

emodpy_malaria.malaria_vector_species_params

module, 81

emodpy_malaria.reporters

module, 59

emodpy_malaria.reporters.builtin

module, 59

emodpy_malaria.vector_config

module, 81

F

fmدا_cfg()

(in module

emodpy_malaria.interventions.drug_campaign),

from_csv()

(in module

emodpy_malaria.demographics.MalariaDemographics),
15

from_params()

(in module

emodpy_malaria.demographics.MalariaDemographics),
15

from_pop_csv()

(in module

emodpy_malaria.demographics.MalariaDemographics),
14

from_template_node()

(in module

emodpy_malaria.demographics.MalariaDemographics),
14

G

get_drug_params()

(in module

emodpy_malaria.malaria_config), 79

get_file_from_http()

(in module

emodpy_malaria.malaria_config), 79

get_species_params()

(in module

emodpy_malaria.malaria_config), 80

get_species_params()

(in module

emodpy_malaria.vector_config), 81

I

InputEIR()

(in module

emodpy_malaria.interventions.inpoteir),

35

InputIvermectin()

(in module

emodpy_malaria.interventions.ivermectin),

38

M

MalariaDemographics

(class in

emodpy_malaria.demographics.MalariaDemographics),
13

MalariaDiagnostic()

(in module

emodpy_malaria.interventions.common),

20

MalariaImmunityReport

(class in

emodpy_malaria.reporters.builtin), 75

MalariaPatientJSONReport

(class in

emodpy_malaria.reporters.builtin), 72

MalariaSqlReport

(class in

emodpy_malaria.reporters.builtin), 74

MalariaSummaryReport

(class in

emodpy_malaria.reporters.builtin), 72

MalariaSurveyJSONAnalyzer

(class in

emodpy_malaria.reporters.builtin), 75

module 35

emodpy_malaria, 13

emodpy_malaria.demographics, 13

emodpy_malaria.demographics.MalariaDemographics, 13

emodpy_malaria.interventions, 15

emodpy_malaria.interventions.bednet, 15

emodpy_malaria.interventions.common, 20

emodpy_malaria.interventions.community_health_worker, 23

emodpy_malaria.interventions.diag_survey, 25

emodpy_malaria.interventions.drug, 27

emodpy_malaria.interventions.drug_campaign, 28

emodpy_malaria.interventions.inputeir, 35

emodpy_malaria.interventions.irs, 37

emodpy_malaria.interventions.ivermectin, 38

emodpy_malaria.interventions.mosquitorelease, 39

emodpy_malaria.interventions.outbreak, 40

emodpy_malaria.interventions.outdoorrestkill, 48

emodpy_malaria.interventions.scale_larval_habitats, 49

emodpy_malaria.interventions.spacespraying, 51

emodpy_malaria.interventions.sugartrap, 52

emodpy_malaria.interventions.treatment_seeking, 53

emodpy_malaria.interventions.udbednet, 54

emodpy_malaria.interventions.vaccine, 57

emodpy_malaria.malaria_config, 79

emodpy_malaria.malaria_vector_species_params, 81

emodpy_malaria.reporters, 59

emodpy_malaria.reporters.builtin, 59

emodpy_malaria.vector_config, 81

MosquitoRelease() (in module *emodpy_malaria.interventions.mosquitorelease*), 39

N

new_intervention() (in module *emodpy_malaria.interventions.inputeir*), 35

new_intervention_as_file() (in module *emodpy_malaria.interventions.bednet*), 17

new_intervention_as_file() (in module *emodpy_malaria.interventions.drug*), 28

new_intervention_as_file() (in module *emodpy_malaria.interventions.inputeir*), 36

new_intervention_as_file() (in module *emodpy_malaria.interventions.irs*), 38

new_intervention_as_file() (in module *emodpy_malaria.interventions.mosquitorelease*), 39

new_intervention_as_file() (in module *emodpy_malaria.interventions.spacespraying*), 51

new_intervention_as_file() (in module *emodpy_malaria.interventions.sugartrap*), 52

new_intervention_as_file() (in module *emodpy_malaria.interventions.udbednet*), 56

new_intervention_as_file() (in module *emodpy_malaria.interventions.vaccine*), 58

P

parameters(*emodpy_malaria.reporters.builtin.MalariaImmunityReport* attribute), 75

parameters(*emodpy_malaria.reporters.builtin.MalariaPatientJSONReport* attribute), 72

parameters(*emodpy_malaria.reporters.builtin.MalariaSqlReport* attribute), 74

parameters(*emodpy_malaria.reporters.builtin.MalariaSummaryReport* attribute), 72

parameters(*emodpy_malaria.reporters.builtin.MalariaSurveyJSONAttribute* attribute), 75

parameters(*emodpy_malaria.reporters.builtin.ReportDrugStatus* attribute), 75

parameters(*emodpy_malaria.reporters.builtin.ReportEventCounter* attribute), 74

parameters(*emodpy_malaria.reporters.builtin.ReportHumanMigration* attribute), 76

parameters(*emodpy_malaria.reporters.builtin.ReportInterventionPopulation* attribute), 78

parameters(*emodpy_malaria.reporters.builtin.ReportMalariaFiltered* attribute), 73

parameters(*emodpy_malaria.reporters.builtin.ReportNodeDemographics* attribute), 76

parameters(*emodpy_malaria.reporters.builtin.ReportNodeDemographics* attribute), 77

parameters(*emodpy_malaria.reporters.builtin.ReportNodeDemographics* attribute), 77

parameters(*emodpy_malaria.reporters.builtin.ReportNodeDemographics* attribute), 77

parameters(*emodpy_malaria.reporters.builtin.ReportSimpleMalariaTransmission* attribute), 73

parameters(*emodpy_malaria.reporters.builtin.ReportVectorGenetics* attribute), 71

parameters (emodpy_malaria.reporters.builtin.ReportVectorMigrationParam() (in module
attribute), 77 emodpy_malaria.malaria_config), 80
parameters (emodpy_malaria.reporters.builtin.ReportVectorStatsSpeciesParam() (in module
attribute), 72 emodpy_malaria.vector_config), 81
parameters (emodpy_malaria.reporters.builtin.ReportVectorStatsMalariaGenetics() (in module
attribute), 78 emodpy_malaria.malaria_config), 79
parameters (emodpy_malaria.reporters.builtin.SpatialReportMalariaFilteredResults() (in module
attribute), 74 emodpy_malaria.vector_config), 81
parameters (emodpy_malaria.reporters.builtin.VectorHabitatReportDrugParams() (in module
attribute), 75 emodpy_malaria.malaria_config), 79
SpaceSpraying() (in module
emodpy_malaria.interventions.spacespraying),
51
ReportDrugStatus (class in
emodpy_malaria.reporters.builtin), 75
ReportEventCounter (class in
emodpy_malaria.reporters.builtin), 74
ReportHumanMigrationTracking (class in
emodpy_malaria.reporters.builtin), 75
ReportInterventionPopAvg (class in
emodpy_malaria.reporters.builtin), 78
ReportMalariaFiltered (class in
emodpy_malaria.reporters.builtin), 73
ReportNodeDemographics (class in
emodpy_malaria.reporters.builtin), 76
ReportNodeDemographicsMalaria (class in
emodpy_malaria.reporters.builtin), 76
ReportNodeDemographicsMalariaGenetics
(class in emodpy_malaria.reporters.builtin), 77
ReportSimpleMalariaTransmissionJSON
(class in emodpy_malaria.reporters.builtin), 72
ReportVectorGenetics (class in
emodpy_malaria.reporters.builtin), 71
ReportVectorMigration (class in
emodpy_malaria.reporters.builtin), 77
ReportVectorStats (class in
emodpy_malaria.reporters.builtin), 71
ReportVectorStatsMalariaGenetics (class in
emodpy_malaria.reporters.builtin), 78
S
set_drug_param() (in module
emodpy_malaria.malaria_config), 79
set_max_larval_capacity() (in module
emodpy_malaria.malaria_config), 80
set_max_larval_capacity() (in module
emodpy_malaria.vector_config), 87
set_parasite_genetics_params() (in module
emodpy_malaria.malaria_config), 79
set_risk_high() (emodpy_malaria.demographics.MalariaDemographics.MalariaDemographics
method), 14
set_risk_lowmedium()
(emodpy_malaria.demographics.MalariaDemographics.MalariaDemographics
method), 13
T
triggered_campaign_delay_event() (in mod-
ule emodpy_malaria.interventions.common),
21
U
UDBednet() (in module
emodpy_malaria.interventions.udbednet),
54
V
VectorHabitatReport (class in
emodpy_malaria.reporters.builtin), 74