
emodpy-malaria

Institute for Disease Modeling

Feb 19, 2021

CONTENTS

1	Installation	3
2	emodpy_malaria	5
2.1	emodpy_malaria package	5
2.1.1	Subpackages	5
2.1.2	Submodules	29
	Python Module Index	31
	Index	33

emodpy-malaria is a collection of Python scripts and utilities created to streamline user interactions with EMOD and idmtools for modeling malaria.

INSTALLATION

You can install emodpy-malaria using the instructions in the GitHub repository README (https://github.com/InstituteforDiseaseModeling/emodpy-malaria/tree/main/examples/start_here).

EMODPY_MALARIA

2.1 emodpy_malaria package

2.1.1 Subpackages

emodpy_malaria.demographics package

Submodules

emodpy_malaria.demographics.MalariaDemographics module

```
class emodpy_malaria.demographics.MalariaDemographics(nodes,
                                                    idref='Gridded
                                                    world
                                                    grump2.5arcmin',
                                                    base_file=None,
                                                    init_prev=0.2)
```

Bases: emod_api.demographics.Demographics.Demographics

This class is derived from emod_api.demographics' Demographics class so that we can set certain defaults for Malaria in construction.

```
emodpy_malaria.demographics.MalariaDemographics.fromBasicNode(lat=0, lon=0,
                                                                pop=1000000.0,
                                                                name=1,
                                                                forced_id=1,
                                                                init_prev=0.2)
```

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

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

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

emodpy_malaria.interventions package

Submodules

emodpy_malaria.interventions.bednet module

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

Simple Bednet with small param set. Note Start_Day is initialized as 1, recommend that this be aligned with the start of the simulation

```
emodpy_malaria.interventions.bednet.BabyBednet(camp, start_day, coverage=1.0, blocking_eff=1, killing_eff=1, repelling_eff=1, usage_eff=1, insecticide=None)
```

BabyBednet is not for babies. It's simpler bednet with just the basic configuration controls.

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

emodpy_malaria.interventions.common module

```
emodpy_malaria.interventions.common.MalariaDiagnostic(camp, Base_Sensitivity, Base_Specificity, Measurement_Sensitivity, Detection_Threshold, Diagnostic_Type, Event_Or_Config='Config')
```

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

emodpy_malaria.interventions.diag_survey module

```

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',
    diagnostic_threshold:
    float = 40,
    measurement_sensitivity:
    float = 0.1, sensitivity: float
    = 1, specificity: float =
    1, event_name:
    str = 'Diagnostic Survey',
    nodeIDs: 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,
    disqualifying_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 **MalariaDiagnos-**

tic class, an individual-level class, to test individuals. Upon positive or negative diagnosis, 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** – `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** – Beginning of the intervention's existence. If triggered, runs on trigger, not on `start_day`.
- **diagnostic_type** – Type of malaria diagnostic used. Default is
- **Available options are** (`**BLOOD_SMEAR**`) –
- **TRUE_INFECTION_STATUS** (*) –
- **BLOOD_SMEAR** (*) –
- **PCR** (*) –
- **PF_HRP2** (*) –
- **TRUE_PARASITE_DENSITY** (*) –
- **HAS_FEVER** (*) –
- **diagnostic_threshold** – The diagnostic detection threshold based on
- ****diagnostic_type**** –
- **BLOOD_SMEAR** – Use the `SusceptibilityMalaria::CheckParasiteCountWithTest()` to get a parasite density to check against the threshold.
- **PCR** – Use the `ReportUtilitiesMalaria::NASBADensityWithUncertainty()` method to calculate a measured parasite density and check against the threshold.
- **PF_HRP2** – Add a new method to get the PfHRP2 value and check against the threshold.
- **TRUE_PARASITE_DENSITY** – Check the true/actual parasite density against the threshold (DEFAULT).
- **HAS_FEVER** – Check the person's fever against the threshold.
- **sensitivity** – Setting for **Base_Sensitivity** in the `MalariaDiagnostic`
- **specificity** – Setting for **Base_Specificity** in the `MalariaDiagnostic`
- **measurement_sensitivity** – setting for **Measurement_Sensitivity** in `MalariaDiagnostic` event_name: Description of the event.
- **nodeIDs** – 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

```
emodpy_malaria.interventions.drug.AntiMalarialDrug(camp, start_day, coverage=1.0,
                                                    drug_name='Chloroquine',
                                                    node_ids=None)
```

AntiMalarialDrug intervention wrapper.

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

emodpy_malaria.interventions.drug_campaign module

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

Add a drug config to the simulation configuration based on its code and add the corresponding AntimalarialDrug intervention to the return dictionary. The drug_code needs to be one identified in the drug_cfg dictionary. For example passing the MDA_ALP drug code, will add the drugs config for Artemether, Lumefantrine, Primaquine to the configuration file and will return a dictionary containing a Full Treatment course for those 3 drugs. :param

drug_code: Code of the drug to add :param drug_ineligibility_duration: used as a flag, if anything is present, we add ["DrugStatus:RecentDrug"] :param as Disqualifying_Properties:

Returns A dictionary containing the parameters for an intervention using the given drug

```

emodpy_malaria.interventions.drug_campaign.add_drug_campaign(camp,      camp-
                                                                paigntype:
                                                                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, diag-
                                                                nostic_sensitivity:
                                                                float = 1, diag-
                                                                nostic_specificity:
                                                                float = 1, 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, nodeIDs:
                                                                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=1Received_Campaig

```

Add a drug intervention, as specified in **campaign_type**, to the campaign. This intervention uses the **MalariaDiagnostic** class to create either a scheduled or a triggered event to the campaign and the **AntimalarialDrug** class to configure drug parameters. You can also specify a delay period for a triggered event that broadcasts afterwards.

Note: Dosing/Dosing_Type was removed, please use `adeherent_drug_configs` if you want something besides a full treatment. Also, `drug_ineligibility_duration` does not prevent diagnostics from being distributed, but it does prevent positively-tested patients from receiving the drugs.

Parameters

- **cb** – The `DTKConfigBuilder` object for building, modifying, and writing campaign configuration files.
- **campaign_type** – Type of drug campaign. Default is **MDA**. Available options are: * **MDA** * **MSAT** * **SMC** * **fMDA** * **MTAT** * **rfMSAT** * **rfMDA**
- **drug_code** – The drug code of the drug regimen (AL, DP, etc; allowable types are defined in `malaria_drugs.py`).
- **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 then the first entry of `start_days` is the day to start listening for the trigger(s).
- **coverage** – Demographic coverage of the distribution (fraction of people at home during campaign).
- **repetitions** – Number of repetitions.
- **tsteps_btwn_repetitions** – Timesteps between the repetitions.
- **diagnostic_type** – Diagnostic config for diagnostic-dependent campaigns: * **MSAT** * **fMDA** * **rfMSAT**
- **diagnostic_threshold** – Diagnostic config for diagnostic-dependent campaigns: * **MSAT** * **fMDA** * **rfMSAT**
- **diagnostic_sensitivity** – Setting for **Base_Sensitivity** in the `MalariaDiagnostic`
- **diagnostic_specificity** – Setting for **Base_Specificity** in the `MalariaDiagnostic`
- **measurement_sensitivity** – setting for **Measurement_Sensitivity** in `MalariaDiagnostic`
- **fmda_radius** – Radius (in km) of focal response upon finding infection. Default is 0. Used with household only.
- **node_selection_type** – Node selection type for broadcasting focal response trigger. Available options are: * **DISTANCE_ONLY**: It will send the event to nodes that are within a given distance. * **MIGRATION_NODES_ONLY**: It will only send the event to nodes that the individual can migrate to. * **DISTANCE_AND_MIGRATION**: It will only send the event to migratable nodes that are within a given distance. Migratable nodes are Local and Regional.
- **trigger_coverage** – Used with RCD (Reactive Case Detection). Fraction of trigger events that will trigger an RCD. Coverage param sets the fraction of individuals reached during RCD response.
- **snowballs** – Number of snowball levels in reactive response.

- **treatment_delay** – For MSAT and fMDA, the length of time between administering diagnostic and giving drugs; for RCD, the length of time between treating index case and triggering 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.
- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of `NodeSetAll`.
- **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** – When set to > 0 , use `IndividualProperties` to prevent people from receiving drugs too frequently. Demographics file will need to define the IP `DrugStatus` with possible values `None` and `RecentDrug`. Individuals with status `RecentDrug` will not receive drugs during drug campaigns, though they are still eligible for receiving diagnostics (in MSAT, etc). Individuals who receive drugs during campaigns will have their `DrugStatus` changed to `RecentDrug` for `drug_ineligibility_duration` days.
- **node_property_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.
- **ind_property_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.
- **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 begin a triggerable drug campaign, such as MDA, MSAT, SMC, fMDA, and MTAT.
- **listening_duration** – Duration to listen for the trigger. Default is -1, which listens indefinitely.
- **adherent_drug_configs** – List of adherent drug configurations, which are dictionaries from `configure_adherent_drug`.
- **target_residents_only** – When set to true (1), the intervention is only distributed to individuals for whom the node is their home node. They are not visitors from another node.
- **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.
- **receiving_drugs_event_name** – Event to send out when person received drugs. Default: 'Received_Campaign_Drugs'

Returns 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, nodeIDs: 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 MDA (mass drug administration) drug intervention to campaign. If there are multiple start days in a list and `trigger_condition_list` is empty, then mda's are created to run on the days in the `start_days` list. If the `triggerer_condition_list` is present, then a triggered mda is created and uses the first day of the `start_days`. If there are repetitions or a `triggered_campaign_delay` then separate nodeleveltriggered interventions are created with a delay that sends out an event that triggers the mda. Multiple start days are only valid for non-triggered mdas.

Parameters

- **camp** – The DTKConfigBuilder
- **for building, modifying, and writing campaign configuration files.** (*object*) –
- **start_days** – List of integers.
- **coverage** – Demographic coverage of mda's.
- **drug_configs** – List of dictionaries of drug configurations to be given out, created in `add_drug_campaign`.
- **receiving_drugs_event** – (Optional) Broadcast event container with event to be broadcast when drugs received.
- **repetitions** – Number of repetitions for mda. For triggered mda, this is for a repeated mda after a trigger.
- **tsteps_btwn_repetitions** – Timesteps between repeated scheduled mdas or between once-triggered repeated mdas.
- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of `NodeSetAll`.
- **expire_recent_drugs** – `PropertyValueChanger` intervention that updates `DrugStatus:Recent drug` to individual properties.

- **node_property_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"}].
- **target_group** – A dictionary targeting an age range and gender of individuals for treatment. In the format {"agemin": x, "agemax": y, "gender": z}. Default is Everyone.
- **trigger_condition_list** – List of event triggers upon which mda(s) are distributed.
- **listening_duration** – Duration to listen for the trigger. Default is -1, which listens indefinitely.
- **triggered_campaign_delay** – Delay period between the trigger and the mda. Default is 0.
- **target_residents_only** – When set to true (1), the intervention is only distributed to individuals for whom the node is their home node. They are not visitors from another node.
- **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

```
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,  
diagnostic_sensitivity: float =  
1, diagnostic_specificity: float =  
1, measurement_sensitivity:  
float = 0.1, nodeIDs: 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', trigger_condition_list: list = None,  
triggered_campaign_delay: int =  
0, listening_duration: int = -  
1, check_eligibility_at_trigger:  
bool = False)
```

Add a MSAT (mass screening and treatment) drug intervention to campaign. This is either scheduled (on days from `start_days`) or triggered (when `trigger_condition_list` is present).

Parameters

- **camp** – The `DTKConfigBuilder` object for building, modifying, and writing campaign configuration files.
- **start_days** – List of days on which to start the intervention.
- **coverage** – Demographic coverage of the intervention.
- **drug_configs** – List of dictionaries of drug configurations to be distributed, created in `add_drug_campaign`.
- **receiving_drugs_event** – (Optional) Broadcast event container with event to be broadcast when drugs received.
- **repetitions** – How many times the intervention will be repeated.
- **tsteps_btwn_repetitions** – Time steps between repetitions.
- **treatment_delay** – Delay before the triggered drug distribution is done.
- **diagnostic_type** – Diagnostic type. Available options are: *
`TRUE_INFECTION_STATUS` * `BLOOD_SMEAR` * `PCR` * `PF_HRP2` *
`TRUE_PARASITE_DENSITY` * `HAS_FEVER`
- **diagnostic_threshold** – Diagnostic threshold values, which are based on the selected diagnostic type.

- **diagnostic_sensitivity** – Setting for **Base_Sensitivity** in the MalariaDiagnostic
- **diagnostic_specificity** – Setting for **Base_Specificity** in the MalariaDiagnostic
- **measurement_sensitivity** – setting for **Measurement_Sensitivity** in MalariaDiagnostic
- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of NodeSetAll.
- **expire_recent_drugs** – PropertyValueChanger intervention that updates DrugStatus to Recent drug in IndividualProperties.
- **node_property_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.
- **ind_property_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.
- **disqualifying_properties** – List of IndividualProperty key:value pairs that cause an intervention to be aborted. For example, [{"IndividualProperty1": "PropertyValue1"}, {"IndividualProperty2": "PropertyValue2"}].
- **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.
- **trigger_condition_list** – List of events to trigger MSAT. When trigger_string is set, the first entry of start_days is the day that is used to start listening for the trigger(s), the campaign happens when the trigger(s) is received.
- **triggered_campaign_delay** – How long to delay the actual intervention (drug giving) for after the trigger is received.
- **listening_duration** – Duration to listen for the trigger. Default is -1, which listens 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. Default is false, checks at distribution.

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,  
                                                    diagnostic_sensitivity: float =  
                                                    1, diagnostic_specificity: float =  
                                                    1, measurement_sensitivity:  
                                                    float = 0.1, fmda_radius: int = 0,  
                                                    node_selection_type: str = 'DIS-  
TANCE_ONLY', nodeIDs: 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', trigger  
                                                    condition_list: list = None,  
                                                    listening_duration: int = - 1,  
                                                    triggered_campaign_delay: int  
                                                    = 0, check_eligibility_at_trigger:  
                                                    bool = False)
```

Add a fMDA (focal mass drug administration) drug intervention to campaign. The fMDA is based on results from diagnostic survey, which is either scheduled (on days from *start_days*) or triggered (when *trigger_condition_list* is present).

Parameters

- **camp** – The `DTKConfigBuilder` object for building, modifying, and writing campaign configuration files.
- **start_days** – List of days on which to start the intervention.
- **trigger_coverage** – Demographic coverage of the triggered intervention.
- **coverage** – Demographic coverage of the intervention.
- **drug_configs** – List of dictionaries of drug configurations to be distributed, created in `add_drug_campaign`.
- **receiving_drugs_event** – (Optional) Broadcast event container with event to be broadcast when drugs received.
- **repetitions** – How many times the intervention will be repeated.
- **tsteps_btwn_repetitions** – Time steps between repetitions.
- **treatment_delay** – Delay before the triggered drug distribution is done.

- **diagnostic_type** – Diagnostic type. Available options are: * TRUE_INFECTION_STATUS * BLOOD_SMEAR * PCR * PF_HRP2 * TRUE_PARASITE_DENSITY * HAS_FEVER
- **diagnostic_threshold** – Diagnostic threshold values, which are based on the selected diagnostic type.
- **diagnostic_sensitivity** – Setting for **Base_Sensitivity** in the MalariaDiagnostic
- **diagnostic_specificity** – Setting for **Base_Specificity** in the MalariaDiagnostic
- **measurement_sensitivity** – setting for **Measurement_Sensitivity** in MalariaDiagnostic
- **fmda_radius** – Radius of the follow up BroadcastToOtherNodes interventions, uses node_selection_type.
- **node_selection_type** – Node selection type for broadcasting fMDA trigger. Available options are: * DISTANCE_ONLY: Nodes located within the distance specified by fmda_type are selected. * MIGRATION_NODES_ONLY: Nodes that are local or regional are selected. * DISTANCE_AND_MIGRATION: Nodes are selected using DISTANCE_ONLY and MIGRATION_NODES_ONLY criteria.
- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of NodeSetAll.
- **expire_recent_drugs** – PropertyValueChanger intervention that updates DrugStatus to Recent drug in IndividualProperties.
- **node_property_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.
- **ind_property_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.
- **disqualifying_properties** – List of IndividualProperty key:value pairs that cause an intervention to be aborted. For example, [{"IndividualProperty1": "PropertyValue1"}, {"IndividualProperty2": "PropertyValue2"}].
- **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.
- **trigger_condition_list** – List of events to trigger fMDA. When trigger_string is set, the first entry of start_days is the day that is used to start listening for the trigger(s), the campaign happens when the trigger(s) is received.
- **listening_duration** – Duration to listen for the trigger. Default is -1, which listens indefinitely.
- **triggered_campaign_delay** – How long to delay the actual intervention (drug giving) after the trigger is received.
- **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. Default is false, checks at distribution.

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, diagnos-  
tic_sensitivity: float = 1,  
diagnostic_specificity: float =  
1, measurement_sensitivity:  
float = 0.1, fmda_radius: int  
= 0, node_selection_type: str  
= 'DISTANCE_ONLY', snow-  
balls: int = 0, nodeIDs: list  
= None, expire_recent_drugs:  
emod_api.interventions.common.PropertyValueChange  
= None,  
node_property_restrictions:  
list = None,  
ind_property_restrictions:  
list = None, disqualify-  
ing_properties: list = None)
```

Add a rfMDA (reactive focal mass drug administration) drug intervention to campaign. Detecting malaria triggers diagnostic surveys to run on neighboring nodes and so on, up to the number of triggered interventions defined in **snowballs** parameter.

Parameters

- **camp** – The `DTKConfigBuilder` object for building, modifying, and writing campaign configuration files.
- **start_day** – List of days on which to start the intervention.
- **coverage** – Demographic coverage of the intervention.
- **drug_configs** – List of dictionaries of drug configurations to be distributed, created in `add_drug_campaign`.
- **receiving_drugs_event** – (Optional) Broadcast event container with event to be broadcast when drugs received.
- **listening_duration** – Duration of the existence of the intervention.
- **treatment_delay** – delay before the triggered drug distribution is done
- **trigger_coverage** – Demographic coverage for the triggered intervention
- **diagnostic_type** – Diagnostic type. Available options are: *
TRUE_INFECTION_STATUS * BLOOD_SMEAR * PCR * PF_HRP2 *
TRUE_PARASITE_DENSITY * HAS_FEVER
- **diagnostic_threshold** – Diagnostic threshold values, which are based on the selected diagnostic type.

- **diagnostic_sensitivity** – Setting for **Base_Sensitivity** in the MalariaDiagnostic
- **diagnostic_specificity** – Setting for **Base_Specificity** in the MalariaDiagnostic
- **measurement_sensitivity** – setting for **Measurement_Sensitivity** in MalariaDiagnostic
- **fmda_radius** – Radius of the follow up BroadcastToOtherNodes interventions, uses node_selection_type.
- **node_selection_type** – Node selection type for broadcasting fMDA trigger. Available options are: * **DISTANCE_ONLY**: Nodes located within the distance specified by fmda_type are selected. * **MIGRATION_NODES_ONLY**: Nodes that are local or regional are selected. * **DISTANCE_AND_MIGRATION**: Nodes are selected using **DISTANCE_ONLY** and **MIGRATION_NODES_ONLY** criteria.
- **snowballs** – Number of triggered interventions after the first.
- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of NodeSetAll.
- **expire_recent_drugs** – PropertyValueChanger intervention that updates DrugStatus to Recent drug in IndividualProperties.
- **node_property_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.
- **ind_property_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.
- **disqualifying_properties** – List of IndividualProperty key:value pairs that cause an intervention to be aborted. For example, [{"IndividualProperty1": "PropertyValue1"}, {"IndividualProperty2": "PropertyValue2"}].

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',  
nodeIDs: 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)
```

This function adds two interventions to your campaign file: 1) “Received_Treatment”- triggered BroadcastEventToOtherNodes of “Give_Drugs_rfMDA” event to fmda_radius, with a “treatment_delay” option, and “coverage” coverage. 2) “Give_Drugs_rfMDA” event-triggered MultiIntervention event distributing “drug_configs” drugs, with “trigger_coverage” demographic coverage, with an option to restrict to {“DrugStatus”: “None”} IndividualProperty and optional receiving_drugs_event.

Upon “Received_Treatment” event a delayed “Give_Drugs_rfMDA” is sent out to (optionally) neighboring nodes, which triggers giving of drugs, and (optionally) an individualproperty change and another event being sent.

Parameters

- **camp** – The DTKConfigBuilder object for building, modifying, and writing campaign configuration files.
- **start_day** – The day the intervention is distributed. Default is 0.
- **coverage** – Demographic coverage of the intervention.
- **drug_configs** – List of dictionaries of drug configurations to be distributed, created in add_drug_campaign.
- **receiving_drugs_event** – Event to be sent out upon receiving drugs. Default is 1, everyone.
- **listening_duration** – Duration of the existence of the intervention. Default is ongoing.
- **treatment_delay** – Delay of treatment (in days) after intervention is triggered.
- **trigger_coverage** – Demographic coverage for intervention triggered by successful treatment.
- **fmda_radius** – Radius (km) of sending event to other nodes.
- **node_selection_type** – Node selection type. Available options are: * DISTANCE_ONLY: Nodes located within the distance specified by fmda_type are selected. * MIGRATION_NODES_ONLY: Nodes that are local or regional are selected. * DISTANCE_AND_MIGRATION: Nodes are selected using DISTANCE_ONLY and MIGRATION_NODES_ONLY criteria.

- **nodeIDs** – The list of nodes to apply this intervention to (**Node_List** parameter). If not provided, set value of NodeSetAll.
- **expire_recent_drugs** – PropertyValueChanger intervention that updates DrugStatus to Recent drug in IndividualProperties.
- **node_property_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.
- **ind_property_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.
- **disqualifying_properties** – list of IndividualProperty key:value pairs that cause an intervention to be aborted. For example, [{"IndividualProperty1": "PropertyValue1"}, {"IndividualProperty2": "PropertyValue2"}].

Returns None

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

By Default, this is within the node-only (Distance_Only with distance=0).

Parameters

- **fmda_type** – Radius in km of the follow up BroadcastToOtherNodes interventions, uses node_selection_type.
- **node_selection_type** – Node selection type for broadcasting to other nodes. Available options are: * DISTANCE_ONLY: Nodes located within the distance specified by fmda_type are selected. * MIGRATION_NODES_ONLY: Nodes that are local or regional are selected. * DISTANCE_AND_MIGRATION: Nodes are selected using DISTANCE_ONLY and MIGRATION_NODES_ONLY criteria.
- **event_trigger** – String that triggers the broadcast.

Returns Configured BroadcastEventToOtherNodes

emodpy_malaria.interventions.irs module

```
emodpy_malaria.interventions.irs.IRSHousingModification(camp, start_day, coverage=1.0, blocking_eff=1, killing_eff=1,
                                                         insecticide=None, node_ids=None)
```

MCV1 Campaign :param coverage: Demographic Coverage :param blocking: :param killing: Note Start_Day is initialized as 1, recommend that this be aligned with the start of the simulation

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

emodpy_malaria.interventions.ivermectin module

```
emodpy_malaria.interventions.ivermectin.Ivermectin(schema_path_container,  
                                                    killing_effect,      start_day=0,  
                                                    target_coverage=1.0,    tar-  
                                                    get_num_individuals=None,  
                                                    killing_duration_box=0,  
                                                    killing_exponential_decay_rate=0)
```

Parameters

- **start_day** – day to give out this ivermectin
- **target_coverage** – probability of choosing an individual
- **target_num_individuals** – number of individuals to choose
- **killing_effect** – initial parasite killing effect
- **killing_duration_box** – box duration for killing effect
- **killing_exponential_decay_rate** – rate at which killing effect decays per day.
Use 0 for box duration only.

Returns: campaign event

emodpy_malaria.interventions.outdoorrestkill module

```
emodpy_malaria.interventions.outdoorrestkill.OutdoorRestKill(schema_path_container,  
                                                                killing_effect,  
                                                                insecti-  
                                                                cide_name=None,  
                                                                start_day=0,    tar-  
                                                                get_coverage=1.0,  
                                                                killing_predecay_duration=0,  
                                                                killing_decay_rate=0,  
                                                                node_ids=None)
```

Parameters

- **schema_path_container** –
- **killing_effect** –
- **insecticide_name** –
- **start_day** –
- **target_coverage** –
- **killing_predcay_duration** –
- **killing_decay_rate** –

Returns:

emodpy_malaria.interventions.spacespraying module

```
emodpy_malaria.interventions.spacespraying.SpaceSpraying(camp, start_day, coverage=1.0, killing_eff=1, insecticide=None, constant_duration=100, node_ids=None)
```

MCV1 Campaign :param coverage: Demographic Coverage :param blocking: :param killing: Note Start_Day is initialized as 1, recommend that this be aligned with the start of the simulation

```
emodpy_malaria.interventions.spacespraying.new_intervention_as_file(camp, start_day, file_name=None)
```

emodpy_malaria.interventions.sugartrap module

```
emodpy_malaria.interventions.sugartrap.SugarTrap(camp, start_day, coverage=1.0, killing_eff=1, insecticide=None, constant_duration=100, node_ids=None)
```

SugarTrap intervention wrapper.

```
emodpy_malaria.interventions.sugartrap.new_intervention_as_file(camp, start_day, file_name=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, cost: int = 5, 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 ex-

clusive. 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"
}
```

- **killing_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 definite 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}
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.

```

- **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'}

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, birht_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.reporters package

Submodules

emodpy_malaria.reporters.builtin module

```
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

config (config_builder, manifest)

parameters: dict

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

Bases: emodpy.reporters.base.BuiltInReporter

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

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

    config (config_builder, manifest)

    parameters: dict

class emodpy_malaria.reporters.builtin.MalariaTransmissionReport (class_name:
                                                                    str = None,
                                                                    parame-
                                                                    ters: dict =
                                                                    <factory>,
                                                                    Enabled:
                                                                    bool = True,
                                                                    Pretty_Format:
                                                                    bool = True)

    Bases: emodpy.reporters.base.BuiltInReporter

    config (config_builder, manifest)

    parameters: dict

```

2.1.2 Submodules

emodpy_malaria.config module

```

emodpy_malaria.config.get_file_from_http(url)
    Get data files from simple http server.

emodpy_malaria.config.set_team_defaults(config, mani)
    Set configuration defaults using team-wide values, including drugs and vector species.

emodpy_malaria.config.set_team_vs_params(config, mani)

emodpy_malaria.config.get_species_params(cb, species)

emodpy_malaria.config.set_team_drug_params(config, mani)

emodpy_malaria.config.get_drug_params(cb, drug_name)

emodpy_malaria.config.set_species(config, species_to_select)
    Use this function to specify which mosquito species to use in the simulation.

```

Parameters

- **config** – schema-backed config smart dict
- **species_to_list** – list of 1 or more strings.

Returns

```

emodpy_malaria.config.set_resistances(config)
    Use this function after you're done calling add_resistance. config is the input and the output

```

`emodpy_malaria.config.add_alleles` (*allele_names_in, allele_inits_in*)

This is public API function for user to add alleles. User specifies the list of alleles and corresponding initial distribution.

`emodpy_malaria.config.add_mutation` (*from_allele, to_allele, rate*)

Public API function for user to add mutations as part of vector genetics configuration. A mutation is specified with a source allele, a destination allele, and a rate

`emodpy_malaria.config.add_trait` (*manifest, sex_genes, allele_pair, trait_name, trait_value*)

Use this function to add traits as part of vector genetics configuration. Should produce something like:

```
{
  "Allele_Combinations": [ ["X", "X"], ["a0", "a1"] ],
  "Trait_Modifiers": { "INFECTED_BY_HUMAN": 0 }
},
```

`emodpy_malaria.config.add_resistance` (*manifest, insecticide_name, species, combo, blocking=1.0, killing=1.0*)

Use this function to add insecticide resistances. An insecticide can have a list of resistances. Add each resistance separately with the same name:

```
Insecticides = [
{
  "Name": "pyrethroid",
  "Resistances": [
    {
      "Allele_Combinations": [
        [
          "a1",
          "a1"
        ]
      ],
      "Blocking_Modifier": 1.0,
      "Killing_Modifier": pyrethroid_killing,
      "Species": "gambiae"
    }
  ]
},
```

`emodpy_malaria.config.set_genetics` (*vsp, manifest*)

Don't need to pass these anymore since they are module variables. But actually need to try with more than one set and see where I end up in terms of design.

PYTHON MODULE INDEX

e

- `emodpy_malaria`, [5](#)
- `emodpy_malaria.config`, [29](#)
- `emodpy_malaria.demographics`, [5](#)
- `emodpy_malaria.demographics.MalariaDemographics`,
[5](#)
- `emodpy_malaria.interventions`, [6](#)
- `emodpy_malaria.interventions.bednet`, [6](#)
- `emodpy_malaria.interventions.common`, [6](#)
- `emodpy_malaria.interventions.diag_survey`,
[7](#)
- `emodpy_malaria.interventions.drug`, [9](#)
- `emodpy_malaria.interventions.drug_campaign`,
[9](#)
- `emodpy_malaria.interventions.irs`, [23](#)
- `emodpy_malaria.interventions.ivermectin`,
[24](#)
- `emodpy_malaria.interventions.outdoorrestkill`,
[24](#)
- `emodpy_malaria.interventions.spacespraying`,
[25](#)
- `emodpy_malaria.interventions.sugartrap`,
[25](#)
- `emodpy_malaria.interventions.udbednet`,
[25](#)
- `emodpy_malaria.reporters`, [28](#)
- `emodpy_malaria.reporters.builtin`, [28](#)

INDEX

A

`add_alleles()` (in module *emodpy_malaria.config*), 29
`add_diagnostic_survey()` (in module *emodpy_malaria.interventions.diag_survey*), 7
`add_drug_campaign()` (in module *emodpy_malaria.interventions.drug_campaign*), 10
`add_fMDA()` (in module *emodpy_malaria.interventions.drug_campaign*), 17
`add_MDA()` (in module *emodpy_malaria.interventions.drug_campaign*), 13
`add_MSAT()` (in module *emodpy_malaria.interventions.drug_campaign*), 15
`add_mutation()` (in module *emodpy_malaria.config*), 30
`add_resistance()` (in module *emodpy_malaria.config*), 30
`add_rfMDA()` (in module *emodpy_malaria.interventions.drug_campaign*), 21
`add_rfMSAT()` (in module *emodpy_malaria.interventions.drug_campaign*), 20
`add_trait()` (in module *emodpy_malaria.config*), 30
`AntiMalarialDrug()` (in module *emodpy_malaria.interventions.common*), 6
`AntiMalarialDrug()` (in module *emodpy_malaria.interventions.drug*), 9

B

`BabyBednet()` (in module *emodpy_malaria.interventions.bednet*), 6
`Bednet()` (in module *emodpy_malaria.interventions.bednet*), 6

C

`config()` (*emodpy_malaria.reporters.builtin.MalariaPatientJSONReport*

method), 29

`config()` (*emodpy_malaria.reporters.builtin.MalariaSummaryReport*
method), 28
`config()` (*emodpy_malaria.reporters.builtin.MalariaTransmissionReport*
method), 29
`config()` (*emodpy_malaria.reporters.builtin.ReportVectorGenetics*
method), 28
`config()` (*emodpy_malaria.reporters.builtin.ReportVectorStats*
method), 28

D

`drug_configs_from_code()` (in module *emodpy_malaria.interventions.drug_campaign*), 9

E

emodpy_malaria
 module, 5
emodpy_malaria.config
 module, 29
emodpy_malaria.demographics
 module, 5
emodpy_malaria.demographics.MalariaDemographics
 module, 5
emodpy_malaria.interventions
 module, 6
emodpy_malaria.interventions.bednet
 module, 6
emodpy_malaria.interventions.common
 module, 6
emodpy_malaria.interventions.diag_survey
 module, 7
emodpy_malaria.interventions.drug
 module, 9
emodpy_malaria.interventions.drug_campaign
 module, 9
emodpy_malaria.interventions.irs
 module, 23
emodpy_malaria.interventions.ivermectin
 module, 24
emodpy_malaria.interventions.outdoorrestkill
 module, 24

emodpy_malaria.interventions.spacespraying module, 25

emodpy_malaria.interventions.sugartrap module, 25

emodpy_malaria.interventions.udbednet module, 25

emodpy_malaria.reporters module, 28

emodpy_malaria.reporters.builtin module, 28

F

fmda_cfg() (in module *emodpy_malaria.interventions.drug_campaign*), 23

from_pop_csv() (in module *emodpy_malaria.demographics.MalariaDemographics*), 5

from_synth_pop() (in module *emodpy_malaria.demographics.MalariaDemographics*), 5

fromBasicNode() (in module *emodpy_malaria.demographics.MalariaDemographics*), 5

G

get_drug_params() (in module *emodpy_malaria.config*), 29

get_file_from_http() (in module *emodpy_malaria.config*), 29

get_species_params() (in module *emodpy_malaria.config*), 29

I

IRSHousingModification() (in module *emodpy_malaria.interventions.irs*), 23

Ivermectin() (in module *emodpy_malaria.interventions.ivermectin*), 24

M

MalariaDemographics (class in *emodpy_malaria.demographics.MalariaDemographics*), 5

MalariaDiagnostic() (in module *emodpy_malaria.interventions.common*), 6

MalariaPatientJSONReport (class in *emodpy_malaria.reporters.builtin*), 29

MalariaSummaryReport (class in *emodpy_malaria.reporters.builtin*), 28

MalariaTransmissionReport (class in *emodpy_malaria.reporters.builtin*), 29

module

emodpy_malaria, 5

emodpy_malaria.config, 29

emodpy_malaria.demographics, 5

emodpy_malaria.demographics.MalariaDemographics, 5

emodpy_malaria.interventions, 6

emodpy_malaria.interventions.bednet, 6

emodpy_malaria.interventions.common, 6

emodpy_malaria.interventions.diag_survey, 7

emodpy_malaria.interventions.drug, 9

emodpy_malaria.interventions.drug_campaign, 9

emodpy_malaria.interventions.irs, 23

emodpy_malaria.interventions.ivermectin, 24

emodpy_malaria.interventions.outdoorrestkill, 24

emodpy_malaria.interventions.spacespraying, 25

emodpy_malaria.interventions.sugartrap, 25

emodpy_malaria.interventions.udbednet, 25

emodpy_malaria.reporters, 28

emodpy_malaria.reporters.builtin, 28

N

new_intervention_as_file() (in module *emodpy_malaria.interventions.bednet*), 6

new_intervention_as_file() (in module *emodpy_malaria.interventions.drug*), 9

new_intervention_as_file() (in module *emodpy_malaria.interventions.irs*), 23

new_intervention_as_file() (in module *emodpy_malaria.interventions.spacespraying*), 25

new_intervention_as_file() (in module *emodpy_malaria.interventions.sugartrap*), 25

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

O

OutdoorRestKill() (in module *emodpy_malaria.interventions.outdoorrestkill*), 24

P

parameters(*emodpy_malaria.reporters.builtin.MalariaPatientJSONReport* attribute), 29

parameters(*emodpy_malaria.reporters.builtin.MalariaSummaryReport* attribute), 28

parameters (*emodpy_malaria.reporters.builtin.MalariaTransmissionReport*
 attribute), 29

parameters (*emodpy_malaria.reporters.builtin.ReportVectorGenetics*
 attribute), 28

parameters (*emodpy_malaria.reporters.builtin.ReportVectorStats*
 attribute), 28

R

ReportVectorGenetics (class in
emodpy_malaria.reporters.builtin), 28

ReportVectorStats (class in
emodpy_malaria.reporters.builtin), 28

S

set_genetics() (in module
emodpy_malaria.config), 30

set_resistances() (in module
emodpy_malaria.config), 29

set_species() (in module *emodpy_malaria.config*),
 29

set_team_defaults() (in module
emodpy_malaria.config), 29

set_team_drug_params() (in module
emodpy_malaria.config), 29

set_team_vs_params() (in module
emodpy_malaria.config), 29

SpaceSpraying() (in module
emodpy_malaria.interventions.spacespraying),
 25

SugarTrap() (in module
emodpy_malaria.interventions.sugartrap),
 25

U

UDBednet() (in module
emodpy_malaria.interventions.udbednet),
 25