hybrid_SIR_ages.yaml

---
# general information
model_name: hybrid_SIR_ages

model_info:
abstract: 'This model is a simple discrete-time, stochastic, hybrid SIR model (with individuals grouped automatically), with demography based on age groups.'

author: 'Sebastien Picault (sebastien.picault@inra.fr)'

time_info:
# time unit used for all time-related parameters (e.g. rates/durations)
time_unit: 'days'

# duration of one time step (in time units)
delta_t: 1

# definition of modelling paradigm, processes and scales
levels:
population:
desc: 'level of the population'
aggregation_type: 'hybrid'
contains:
- individuals
desc: 'level of the individuals'

individuals:

metapop:
desc: 'level of the metapopulation'
aggregation_type: 'metapopulation'
# The metapopulation is explicitly linked to a specific class in a Python code add-on
file: metapop_movements.py
class_name: Metapopulation

processes:
# now two processes at population level
population:
infection
- aging

---

parameters:
initial_population_size:
value: 100

initial_prevalence:
desc: 'initial proportion of infectious individuals in the population'
value: 0.1

transmission_I:
desc: 'transmission rate from infectious individuals (/day)'

---
hybrid_SIR_ages_metapop.yaml

---
# general information
model_name: hybrid_SIR_ages_metapop

model_info:
abstract: 'This model is a simple discrete-time, stochastic, hybrid SIR model (with individuals grouped automatically), with demography based on age groups, at the metapopulation scale, with data-based trade movements.'

author: 'Sebastien Picault (sebastien.picault@inra.fr)'

time_info:
# time unit used for all time-related parameters (e.g. rates/durations)
time_unit: 'days'

# duration of one time step (in time units)
delta_t: 1

# specify a longer simulation duration than default
total_duration: 200

# specify the date where simulation starts (to connect with recorded data)
origin: 'January 1, 2018'

levels:
population:
desc: 'level of the population'
aggregation_type: 'hybrid'
contains:
- individuals
desc: 'level of the individuals'

metapop:
desc: 'level of the metapopulation'
aggregation_type: 'metapopulation'
# The metapopulation is explicitly linked to a specific class in a Python code add-on
file: metapop_movements.py
class_name: Metapopulation

parameters:
initial_population_size:
value: 100

initial_prevalence:
desc: 'initial proportion of infectious individuals in the population'
value: 0.1

transmission_I:
desc: 'transmission rate from infectious individuals (/day)'

---
### # variables
#### initial_prevalence:
- desc: 'initial proportion of infected populations'
- value: 0.1

### # prototypes = examples of typical agents for each level,
# characterized by specific variable values
#### - newborn:
  - desc: 'newly created individuals'
  - health_state: default
  - age_group: J

#### - imported_movement:
  - desc: 'profile of individuals from outside the metapopulation, assuming no external risk'
  - health_state: S
  - age_group: default

#### - healthy_pop:
  - desc: 'population initially infection-free'
  - initial_prevalence: 0

#### - infected_pop:
  - desc: 'population initially infected with prevalence 0.1'
  - initial_prevalence: 0.1

### # initial conditions
#### initial_conditions:
#### population:
- prototype: healthy
  - amount: 'initial_population_size * (1 - initial_prevalence)'
- prototype: infected
  - amount: 'initial_population_size * initial_prevalence'