BRICS-STI
EpiModeing Project

A Brief Introduction of LibEpidemic

Jingyuan Wang, Honghao Shi, Qijian Tian

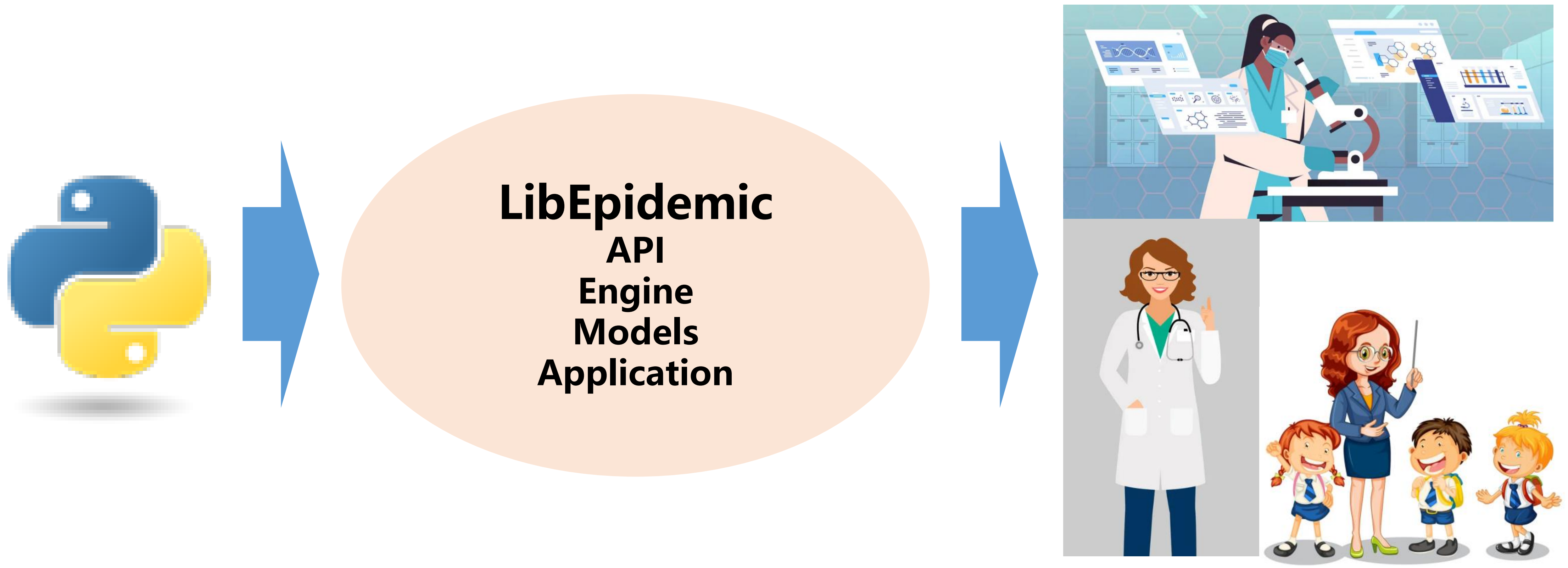
School of Computer Science and Engineering
Beihang University, Beijing, China
Chinese Center for Disease Control and Prevention,
Beijing, China

Contents

CONTENTS

Research Purpose

- LibEpidemic is an open source modeling framework, built in python, designed to help **researchers, epidemiologists, teachers, and students** whose research and learn related to infectious disease modeling.



Target Users - Researcher

- Researchers can use the framework to **build any compartment model simply and quickly**, and execute it with arbitrary parameters to obtain results.

- User Purpose:
  - Build compartment model
  - Execute model
  - Visualization results
- Need to Learn:
  - Python (as basic)
  - API
  - Engine
  - Models
- Can Use:
  - Everything in LibEpidemic
  - Customization at any Level

LibEpidemic
API
Engine
Models
Application

Target Users - Epidemiologist

- Epidemiologists need to use the existing mature infectious disease models, make a small amount of **expansion according to the actual situation**, and then complete the model execution by adjusting the **no-code method** of the configuration file, and obtain the visualization results.

  - User Purpose:
    - Use compartment model
    - Solve Problem in Epidemiology
    - Fine-tune model if necessary
  - Need to Learn:
    - The Principal of Engine (with no code)
    - How to execute python file
  - Can Use:
    - All Existing Models in LibEpidemic
    - API with GUI to fine-tune (in the future)

LibEpidemic
~~API~~
~~Engine~~
Models
Application

Target Users - Educator

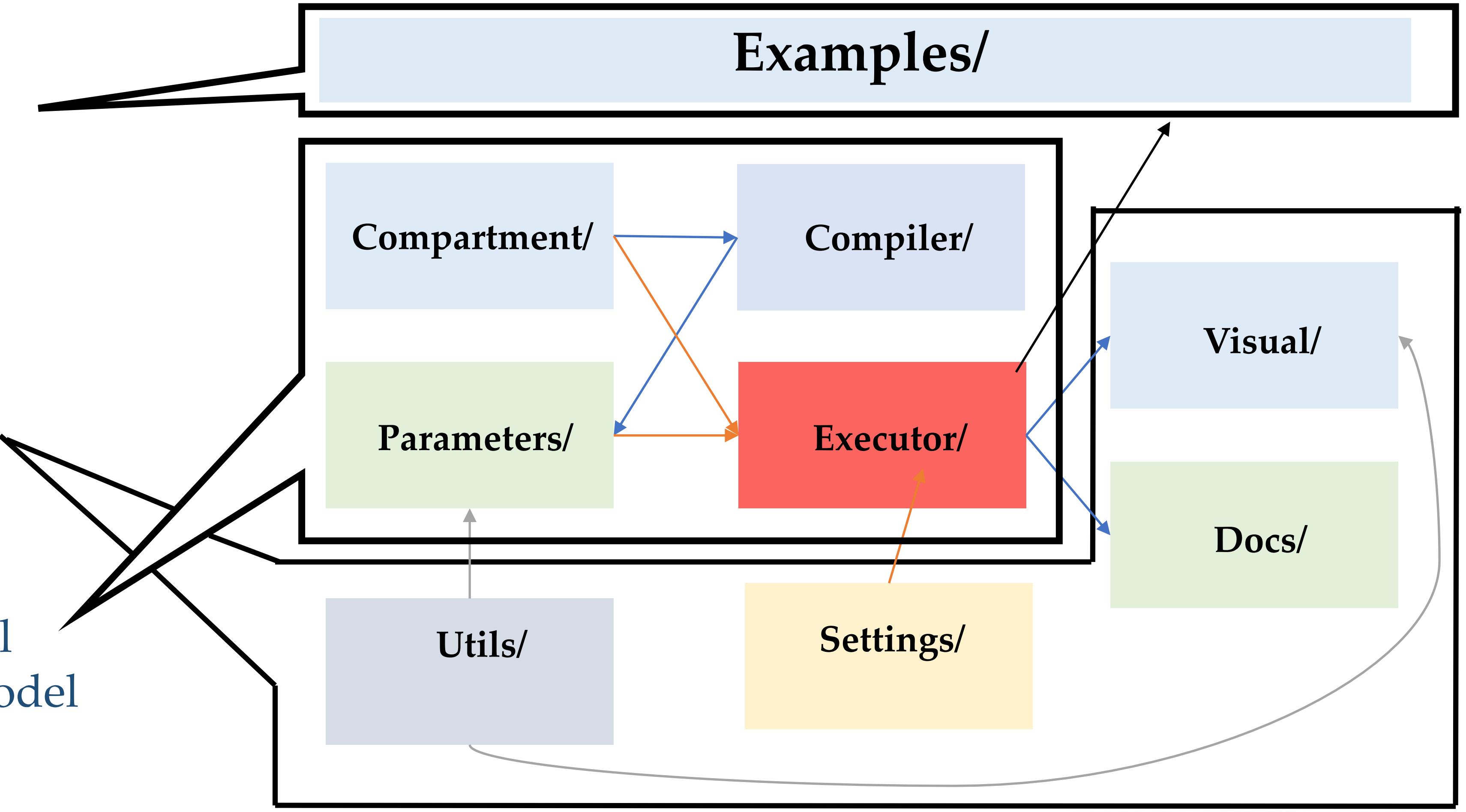
- Researchers The construction and improvement of this framework can promote the integration of teaching and scientific research in related majors. The framework development team has cooperated with the School of Computer Science and Engineering, Beihang University to **carry out the experimental course** “scientific research classroom”, using the framework to carry out the teaching work.

  - User Purpose:
    - Teaching the Principal of Epidemic Modeling
    - Experimental Teaching of Epidemic Modeling
  - Can Use:
    - Learn Epidemic Modeling
    - Learn How to Constuct Model with python
    - Learn How to Improve Model after COVID-19 outbreaking
    - Learn How to Construct Open-source Framework

Overall Structure

- In order to achieve the above goals, LibEpidemic has built an **engine-function-model/application three-level framework**, which provides users with different levels of freedom and support for different purposes.

- Model/Application
  - Basic Model
  - Preset Complex Model
  - NPI applications
- Function
  - Visualization
  - Docs
  - No-code GUI (future)
- Engine
  - API to construct model
  - Executor to execute model
  - Supported tools

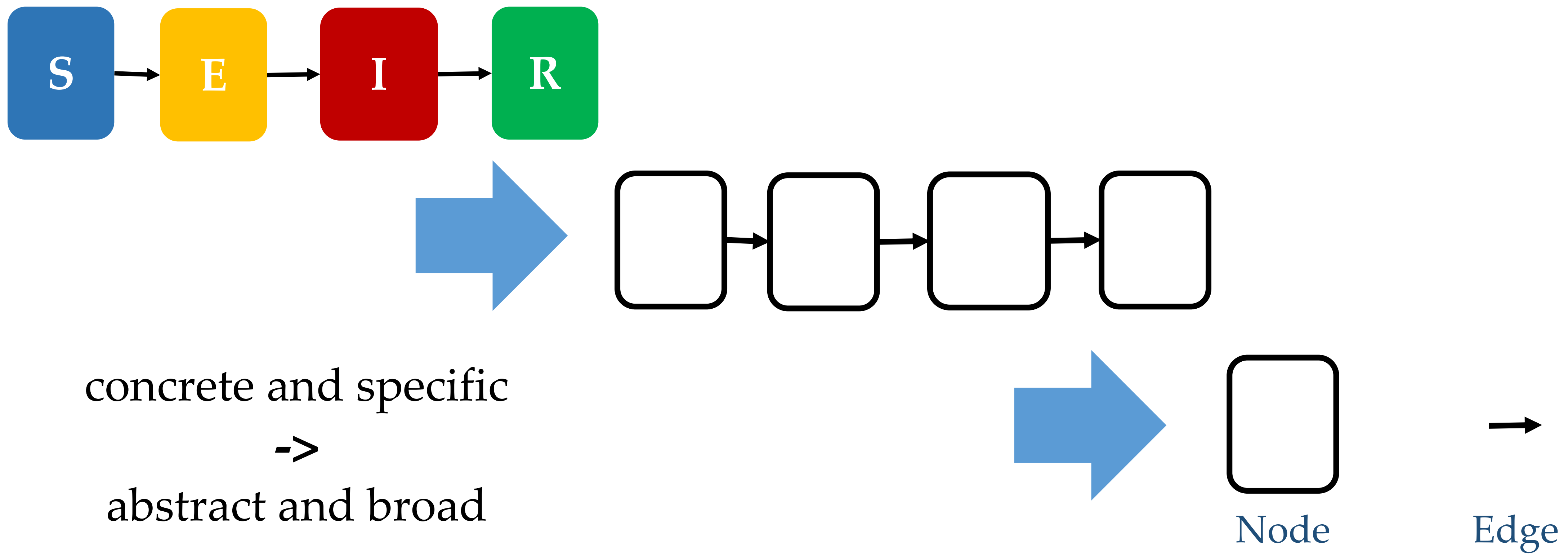


Contents

CONTENTS

Engine - Node and Edge

- In compartment models (dynamic models), the "dynamic" part is the process by which a certain number of individuals move from one group to another. This process can be modeled using a directed graph. Therefore, the bottom layer of this engine is the **nodes and edges of the directed graph**. Each node models a compartment, and each edge models a term of a differential equation.



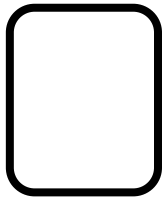
concrete and specific

->

abstract and broad

Engine - Node and Edge (2)

- We use the adjacency list data structure to model and store Node and Edge.



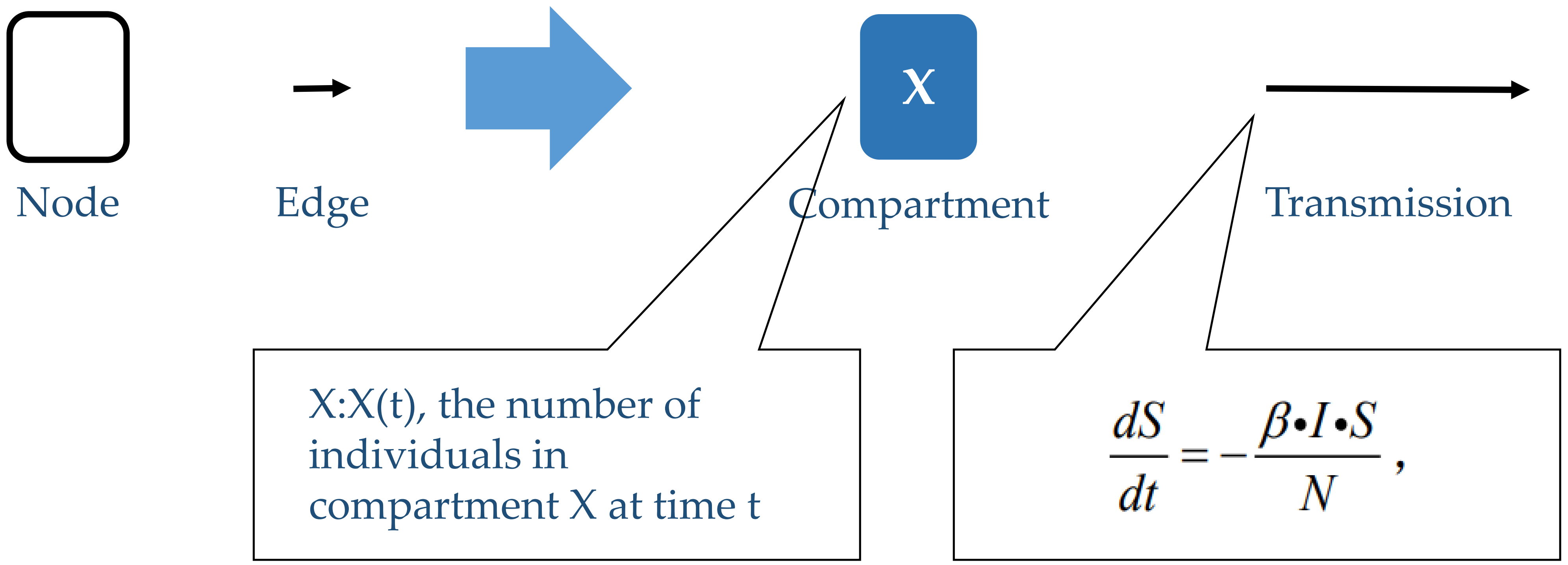
Node

Edge

- Attribute:
  - Name
  - Suffix-list
- Function:
  - Add-suffix
- Example:
  - S->E in SEIR model
    - Init 2 Nodes with Name S and E.
    - Node named S add a suffix E, then the Suffix-list of S is [E].
  - P->A & P -> I in SEPIAR model
    - Init 3 Nodes with Name P, A and I.
    - Node named P add a suffix A, then the Suffix-list of S is [A].
    - Node named P add a suffix I, then the Suffix-list of S is [A, I].

Engine – Compartment and Transmission

- In order to move from node-edge to a real compartment model, a higher-level class containing node-edge and additional information is needed to model the "node" and "edge" in compartment models.
- In the compartment model, the node function is carried by the compartment, and the edge function is carried by the transmission.



X:X(t), the number of individuals in compartment X at time t

$$\frac{dS}{dt} = -\frac{\beta \bullet I \bullet S}{N},$$

Engine – Compartment and Transmission (2)

X

Compartment

- Attribute:
  - Node
  - Value
- Function:
  - Ø
- Example:
  - S in SEIR model
    - Init 4 Nodes with Name S, E, I and R.
    - Construct the graph of Node and Edge. (Detail in next Slide)
    - Init 1 Compartment from Node S.
    - Fill the default value of S with 10000.

Engine – Compartment and Transmission (3)

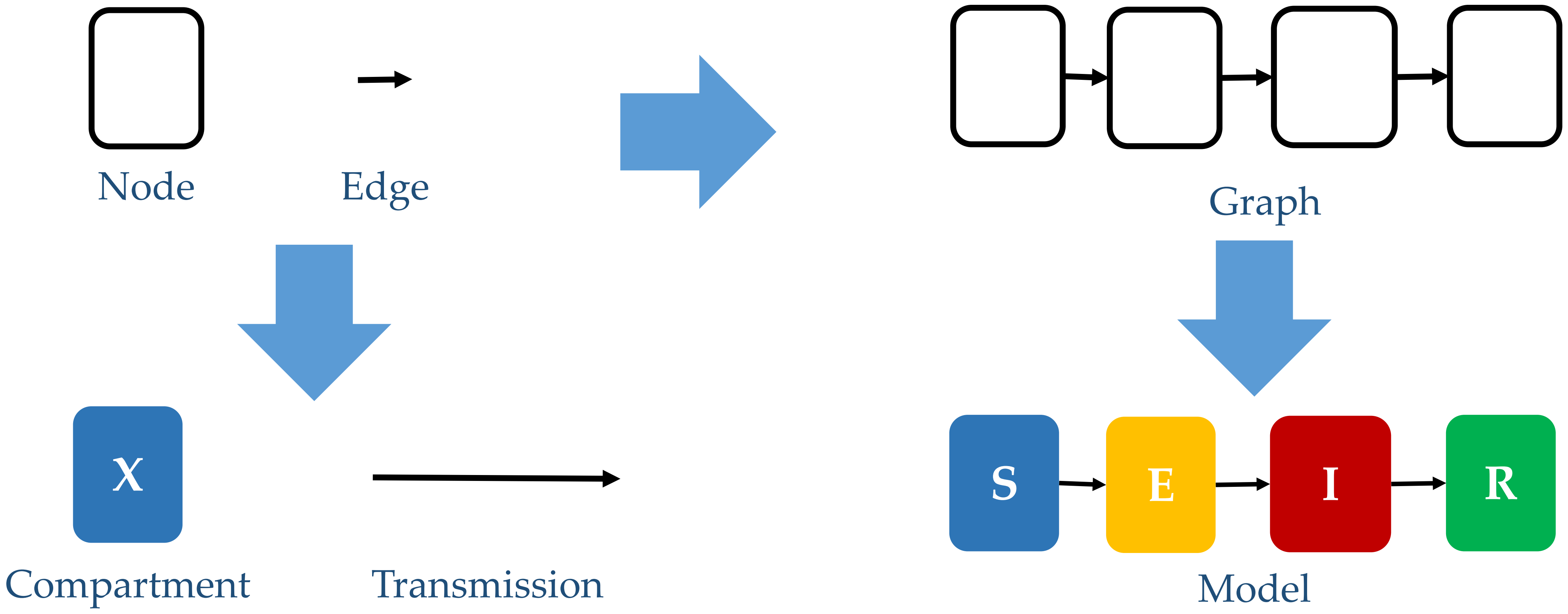
$\longrightarrow$

Transmission

- Attribute:
  - Expression
  - Parameter-values
- Function:
  - Set-Exp
  - Set-Param
- Example:
  - S->E in SEIR model
    - Init Graph, finish Compartment.
    - Set the Expression with Set-Exp: beta * S * I * popuinv
    - Set the Parameter-values with Set-Param: {beta: 0.3}
    - Add the value of popuinv with 1.0/popu, Parameter-values:{beta:0.3, popuinv:0.0001}

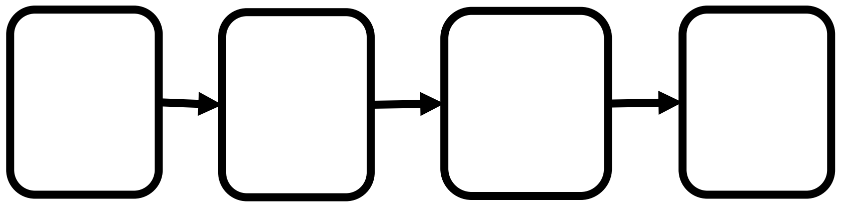
Engine – Graph and Model

- The graph and model **integrate the various parts** of the compartment model. Graph integrates node and path at the level of directed graph, while model integrates compartment and transmission at the level of infectious disease.

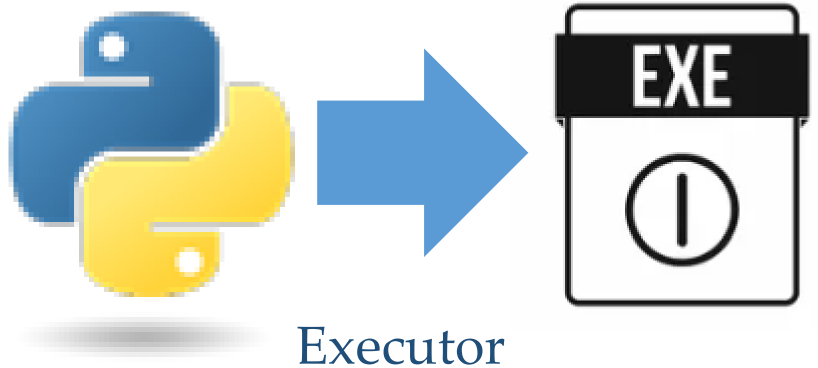


API User Guide – SEIR for Example

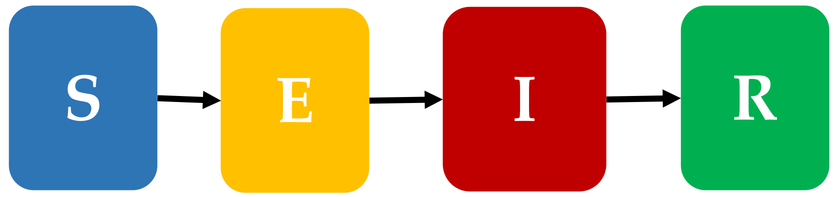
- LibEpidemic provides 4 types of APIs for building a compartment model:
  - Structure
  - Formula and Parameters
  - Executor
  - Results and Visualization
- Here we use the built and simulated SEIR as an example to introduce these APIs



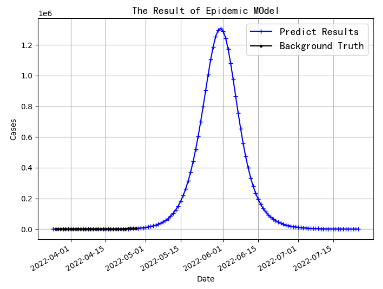
Structure

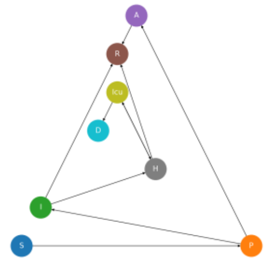


Executor



Formula and Parameters

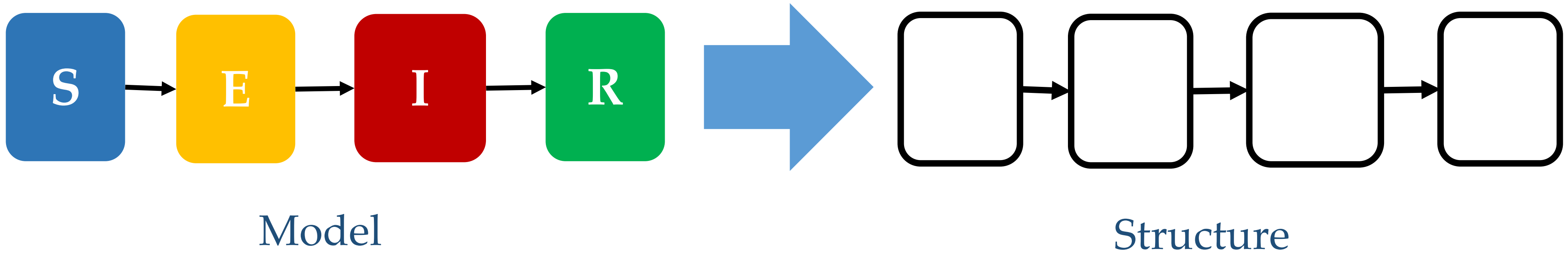




Results and Visualization

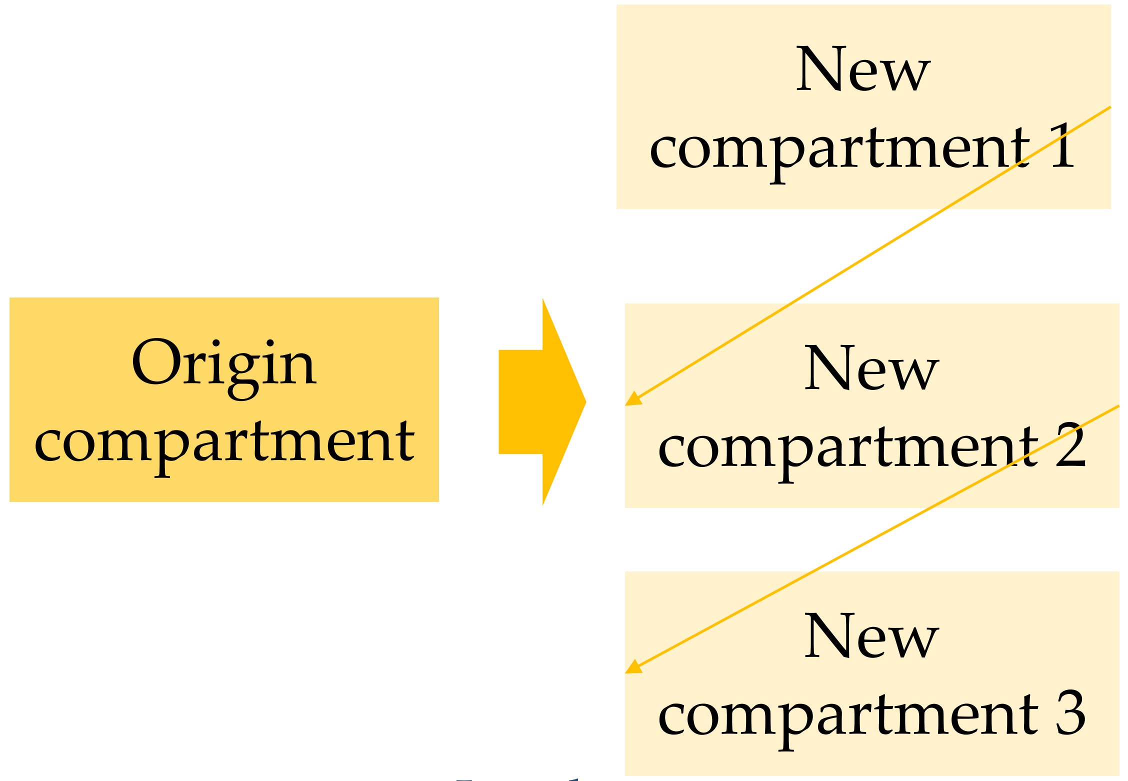
API User Guide – Determine Structure

- You can build a compartment model structure of any shape through the following three APIs:
  - Vertical Divide
  - Horizontal Divide
  - Add Path
- These APIs can not only build a compartment model of arbitrary structure, but also **have specific epidemiological significance**, which is one of the research points in the field of infectious disease modeling after the outbreak of COVID-19

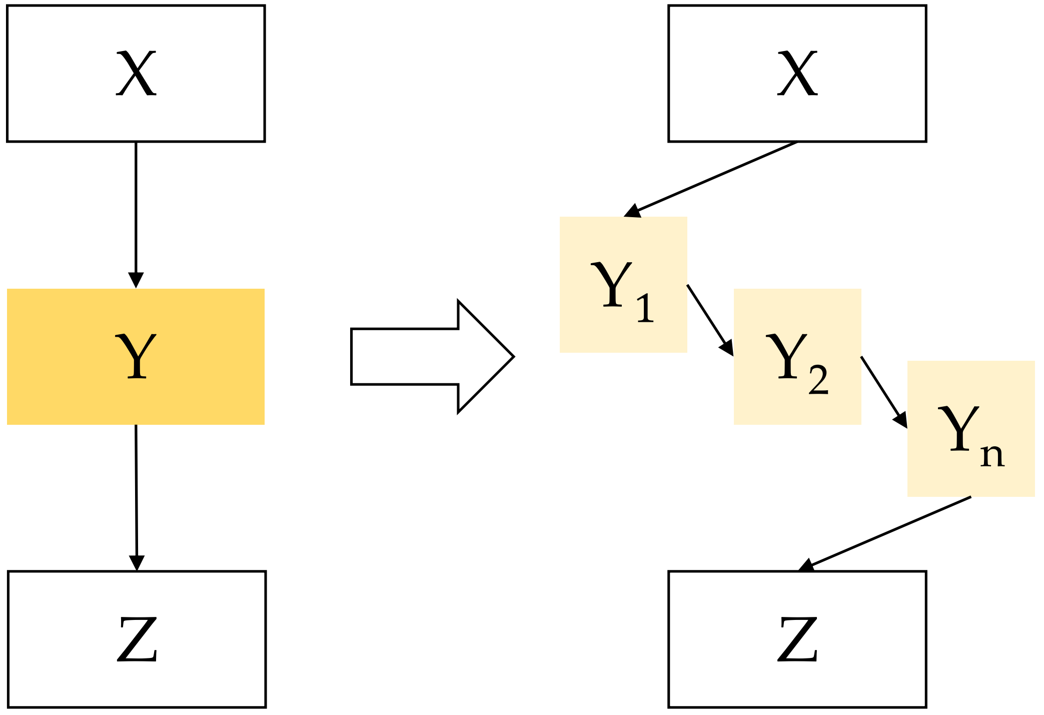


API User Guide – Determine Structure (2)

- Vertical Divide can subdivide compartments by information relevant to the epidemiological process, such as asymptomatic, pre-symptomatic, confirmed, isolated, etc.
- Such subdivide supports the modeling of **individual differences within the same compartment**



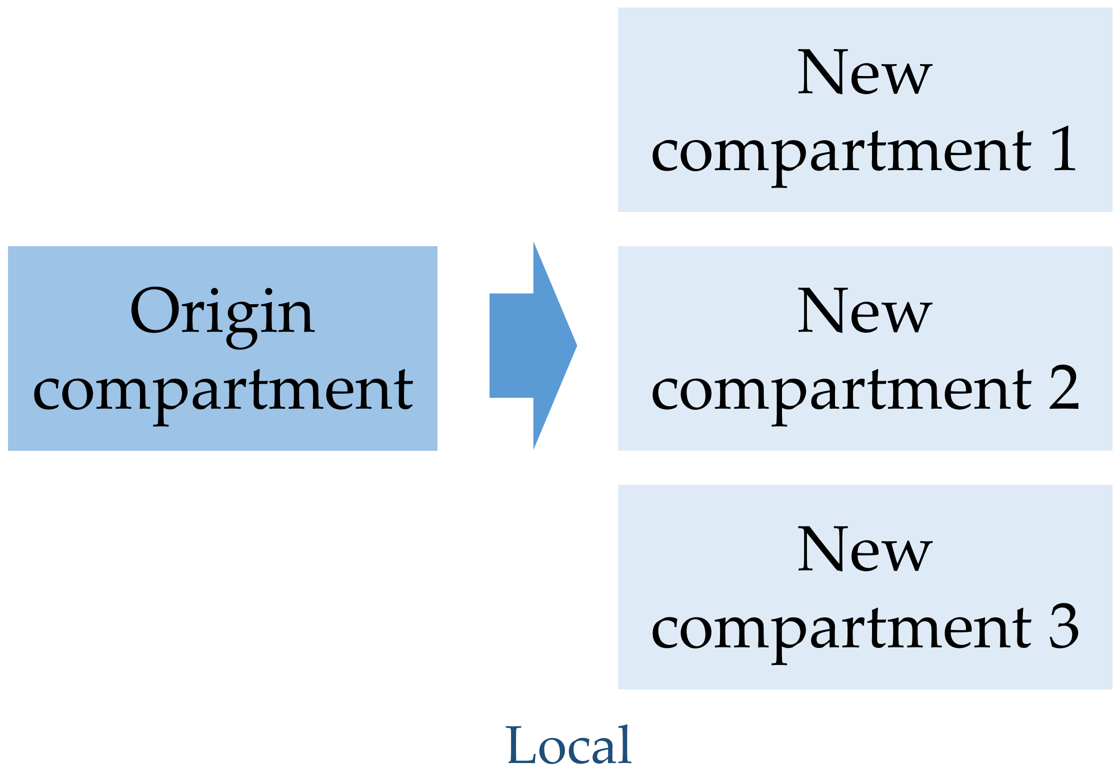
Local

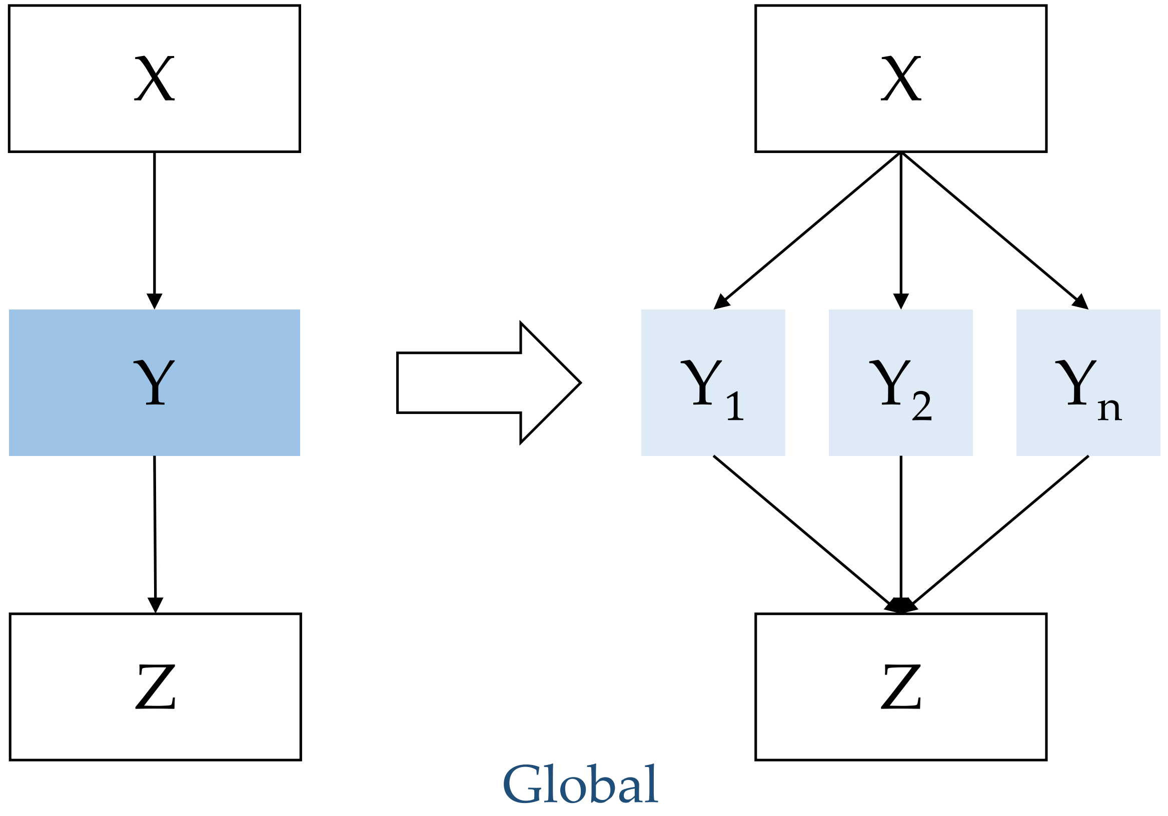


Global

API User Guide – Determine Structure (3)

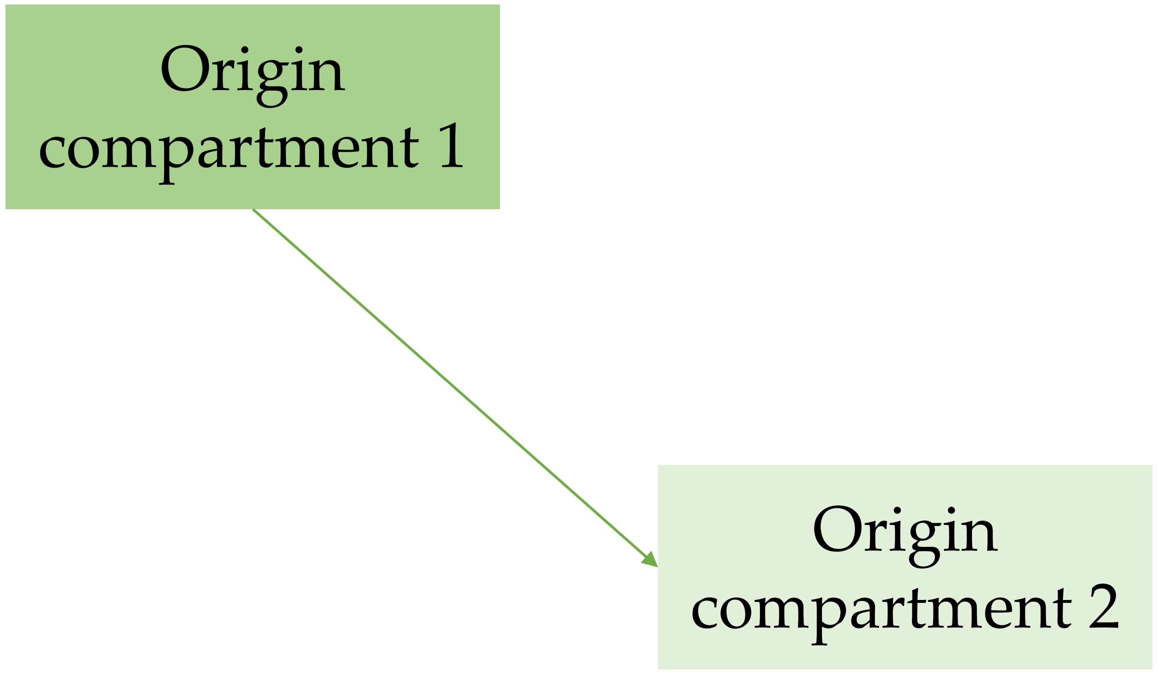
- Horizontal Divide can subdivide compartments by information not relevant to the epidemiological process, such as age, gender, country, etc.
- Such subdivide supports the modeling of **individual differences within the same compartment**



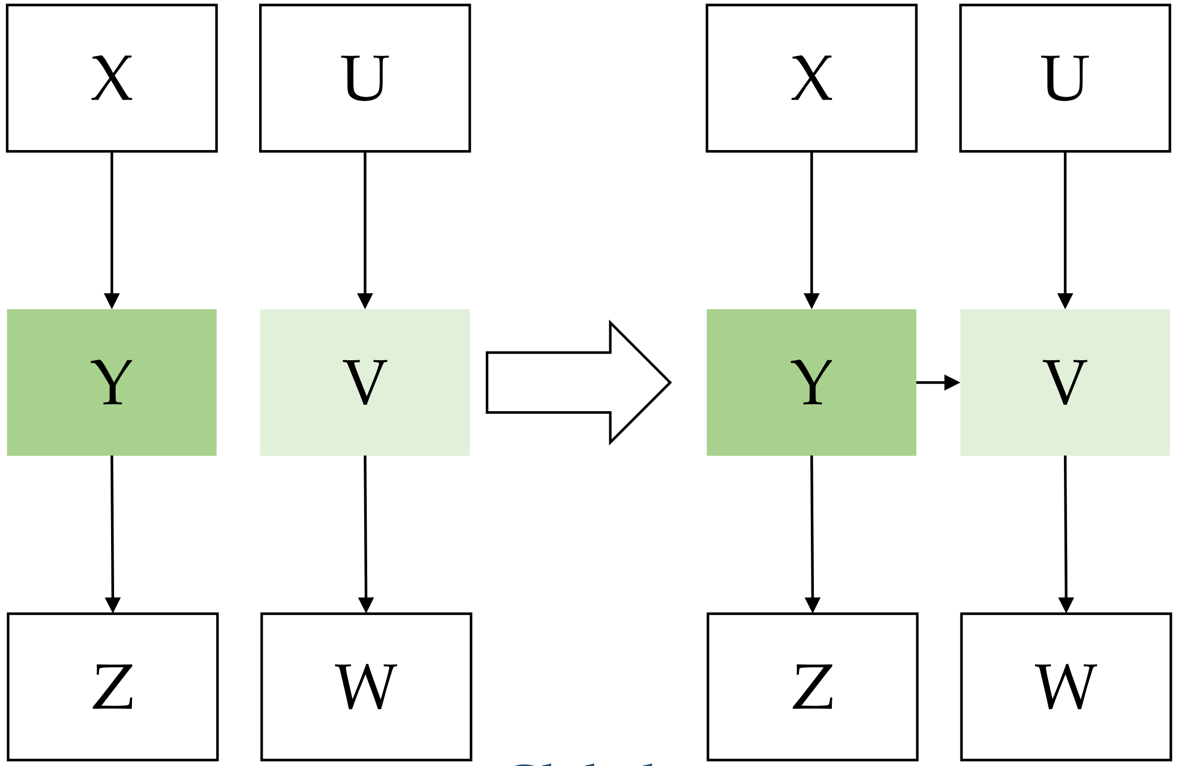


API User Guide – Determine Structure (4)

- Add Path can add an transmission, give it formula and parameters.
- Usually used in combination with the first two methods



Local



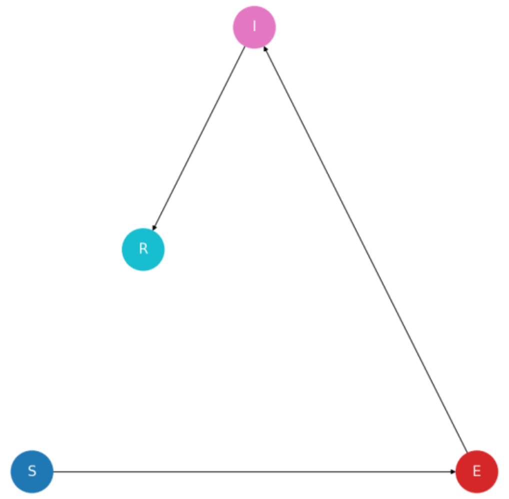
Global

API User Guide – Determine Structure (4)

- Determine the structure of SEIR model with these API:
  - Initializes a graph containing the S compartment and names it basic_SEIR.
  - Uses **vertical_divide** to subdivide the initial S into S, E, I and R, which corresponds to the SEIR model.
  - Converts the diagram used to describe the abstract structure into a model in LibEpidemic after determining the structure of the compartment model, providing support for subsequent work.
  - Calls the visualization API and prints the structure diagram that builds this model.

```
graph = Graph('basic_SEIR', 'S')
vertical_divide(graph, 'S', ['E', 'I', 'R'])
model = Model('basic_SEIR', graph)
visual_model(model)
```

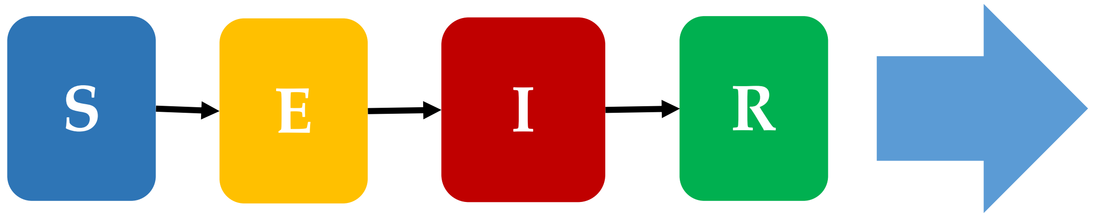
Python Code
in LibEpidemic



Visualization of SEIR
in LibEpidemic

API User Guide – Formula and Parameters

- After create the strcture of model, you can use the following APIs to dealing with parameters(or formula).
  - Set Path Exp
  - Set Path Parameters
  - Reset Parameters
- It is worth noting that **only the parameters in the formula** need to be considered here. The current compartment value involved in the formula and how to calculate the formula are all functions of the executor.



Model

$$\frac{dS}{dt} = -\frac{\beta \bullet I \bullet S}{N},$$

$$\frac{dE}{dt} = \frac{\beta \bullet I \bullet S}{N} + \alpha \bullet E\ ,$$

$$\frac{dI}{dt} = \alpha \bullet E - \gamma \bullet I\ ,$$

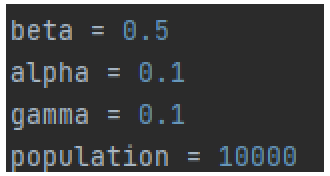
$$\frac{dR}{dt} = \gamma \bullet I\ 。$$

Parameter	Value
β	0.5
α	1.0/10.0
γ	1.0/10.0
N	10000

Formula and Parameters

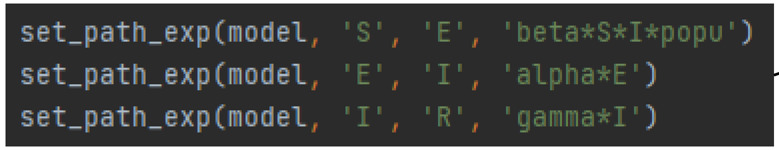
API User Guide – Formula and Parameters (2)

- Determine the Formula and Parameters of SEIR model with these API:
  - Give a set of value



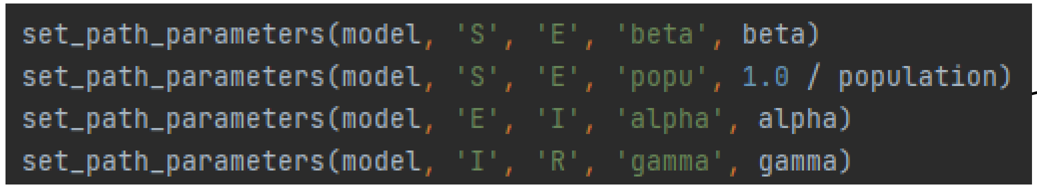
```
beta = 0.5
alpha = 0.1
gamma = 0.1
population = 10000
```

  - Set formula



```
set_path_exp(model, 'S', 'E', 'beta*S*I*popu')
set_path_exp(model, 'E', 'I', 'alpha*E')
set_path_exp(model, 'I', 'R', 'gamma*I')
```

  - Fill Parameters into formula



```
set_path_parameters(model, 'S', 'E', 'beta', beta)
set_path_parameters(model, 'S', 'E', 'popu', 1.0 / population)
set_path_parameters(model, 'E', 'I', 'alpha', alpha)
set_path_parameters(model, 'I', 'R', 'gamma', gamma)
```

$$\frac{dS}{dt} = -\frac{\beta \bullet I \bullet S}{N},$$

$$\frac{dE}{dt} = \frac{\beta \bullet I \bullet S}{N} + \alpha \bullet E \quad ,$$

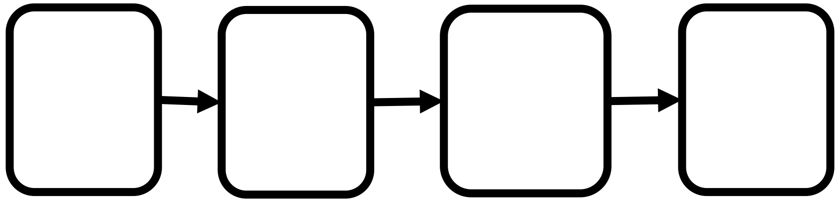
$$\frac{dI}{dt} = \alpha \bullet E - \gamma \bullet I \quad ,$$

$$\frac{dR}{dt} = \gamma \bullet I \quad 。$$

Parameter	**Value**
β	0.5
α	1.0/10.0
γ	1.0/10.0
N	10000

API User Guide – Executor

- Executor can use the model and all the methods of each class, and use the expression analysis system to complete the simulation of the compartment model.
- The function of executor contains:
    - Parse the expression like the string 'beta*S*I*popu'.
    - Calculate the expression to simulate the model.



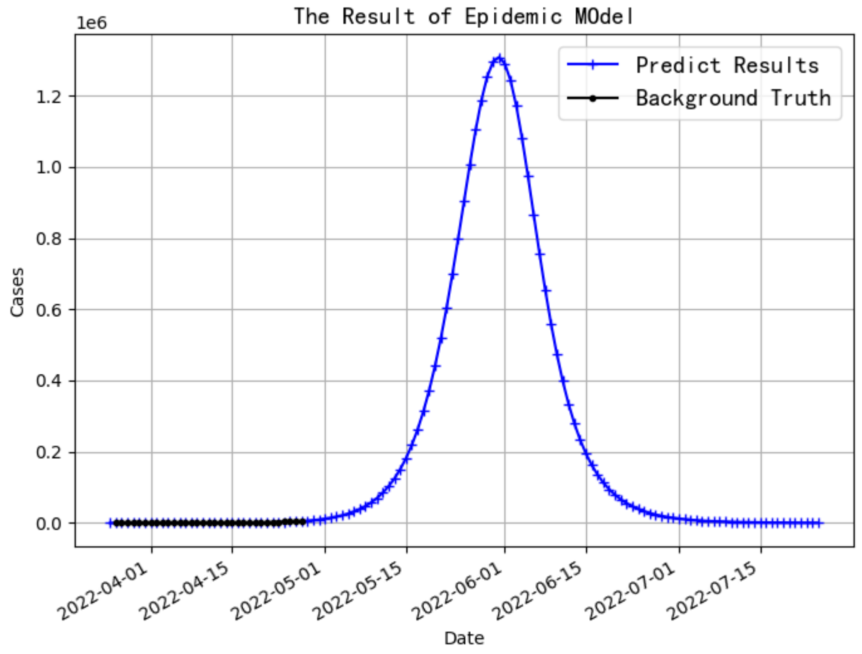
$$\frac{dS}{dt} = -\frac{\beta \cdot I \cdot S}{N},$$

$$\frac{dE}{dt} = \frac{\beta \cdot I \cdot S}{N} + \alpha \cdot E \quad ,$$

$$\frac{dI}{dt} = \alpha \cdot E - \gamma \cdot I \quad ,$$

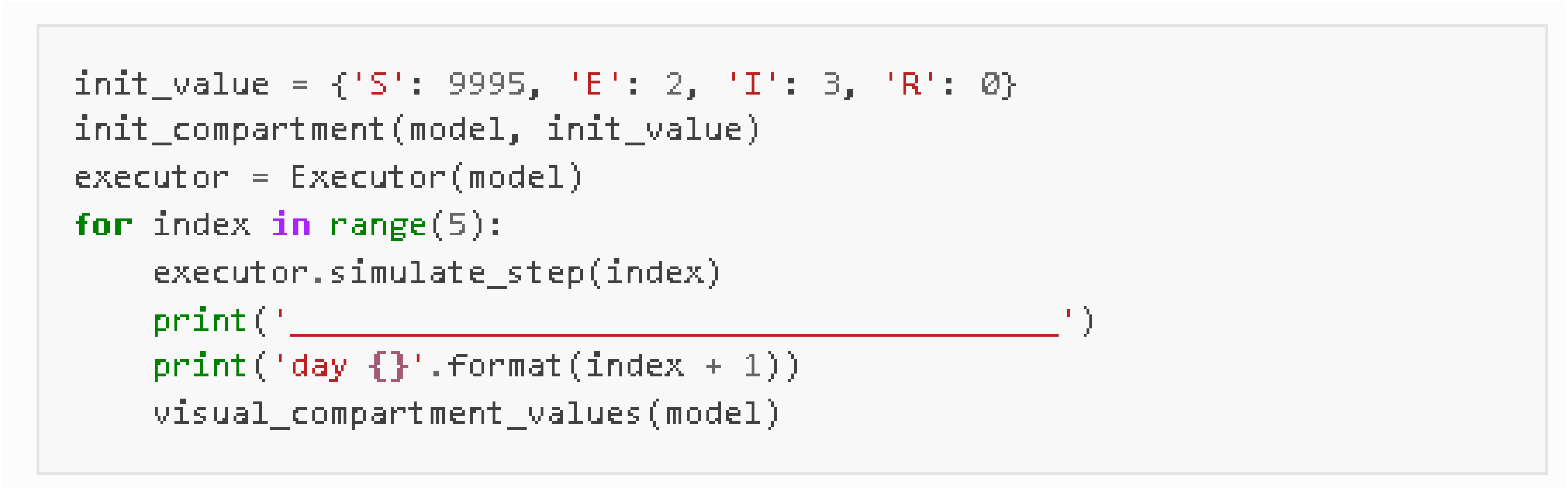
$$\frac{dR}{dt} = \gamma \cdot I \quad 。$$

Parameter	Value
β	0.5
α	1.0/10.0
γ	1.0/10.0
N	10000



API User Guide – Executor (2)

- Executing the SEIR model can be done with only 2 APIs, **without programming the principles** of infectious disease dynamics models, and **without manually writing** complex model extensions. In fact, executing steps is the same to every model, not only SEIR.
  - Use **init_compartment** to set the init value of each compartment.
  - Use Executor() to init a executor
  - Use **simulate_step** to finish all parsing and calculating



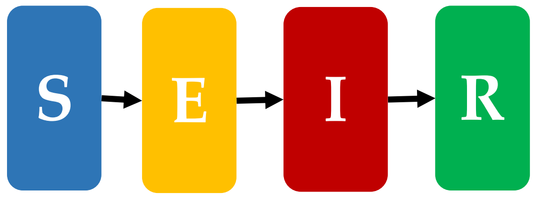
```
init_value = {'S': 9995, 'E': 2, 'I': 3, 'R': 0}
init_compartment(model, init_value)
executor = Executor(model)
for index in range(5):
    executor.simulate_step(index)
    print('_______________________________________')
    print('day {}'.format(index + 1))
    visual_compartment_values(model)
```

API and Engine - Conclusion

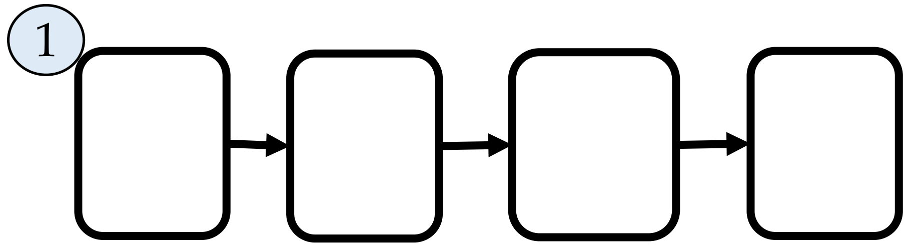
- In LibEpidemic, you can:
  - Build the structure of model with 3 APIs:
    - Vertical Divide
    - Horizontal Divide
    - Add Path
  - Fill with formula and parameters with 3 APIs:
    - Set Path Exp
    - Set Path Parameters
    - Reset Parameters
  - Execute the model with 2 APIs:
    - init_compartment
    - simulate_step
  - Visualizing the results with 4 APIs or matplotlib:
    - visual_graph
    - visual_model
    - visual_compartment_values
    - plot_line
    - **Import matplotlib as plt (*)**

API and Engine – Conclusion (2)

- In LibEpidemic, you can build a model like SEIR in 4 steps:



Model to be built



1. Determine Structure

2

$$\frac{dS}{dt} = -\frac{\beta \bullet I \bullet S}{N},$$

$$\frac{dE}{dt} = \frac{\beta \bullet I \bullet S}{N} + \alpha \bullet E \quad ,$$

$$\frac{dI}{dt} = \alpha \bullet E - \gamma \bullet I \quad ,$$

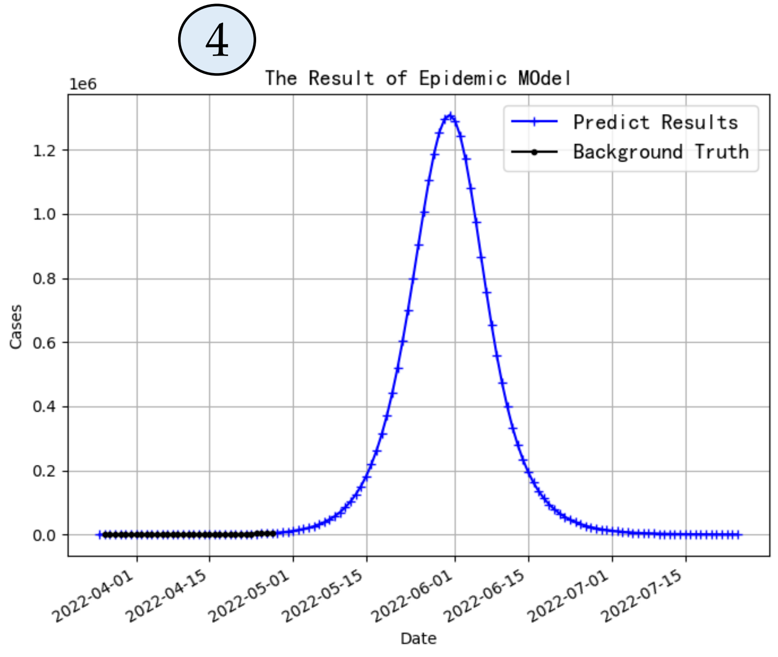
$$\frac{dR}{dt} = \gamma \bullet I \quad 。$$

Parameter	Value
β	0.5
α	1.0/10.0
γ	1.0/10.0
N	10000

2. Fill Formula and Parameters

3

3. Executing



4. Visualization

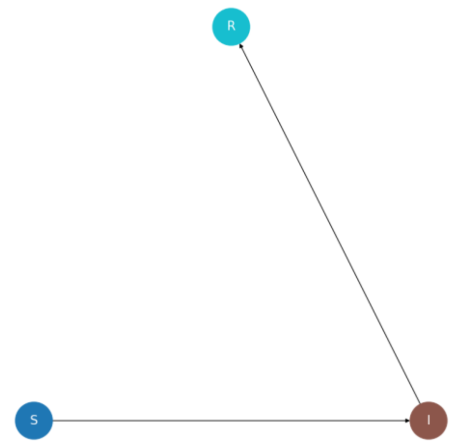
Contents

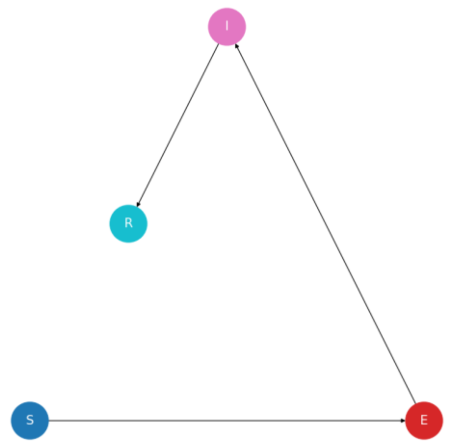
CONTENTS

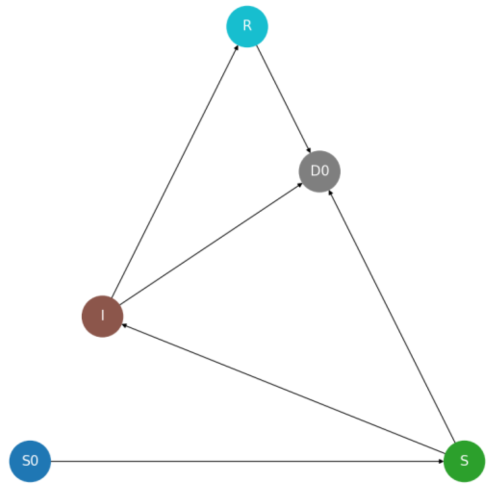
Base Model - SEIR-class

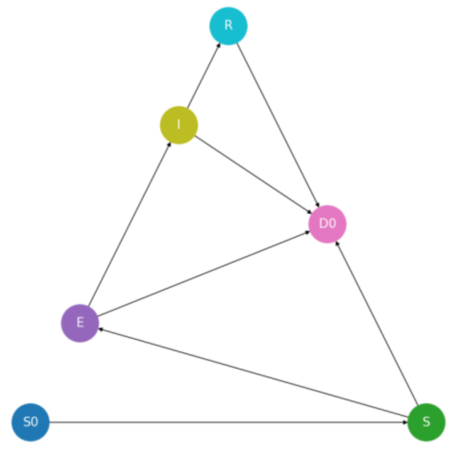
- LibEpidemic has 5 complete basic models built in:
  - SIR
  - SEIR
  - SIR + natural birth/death
  - SEIR + natural birth/death
  - SEPIR

  They come from textbooks or cups for early COVID-19 predictions
- Users can view and modify the code of these basic models, fill in any parameter settings, complete the model simulation, and view the results



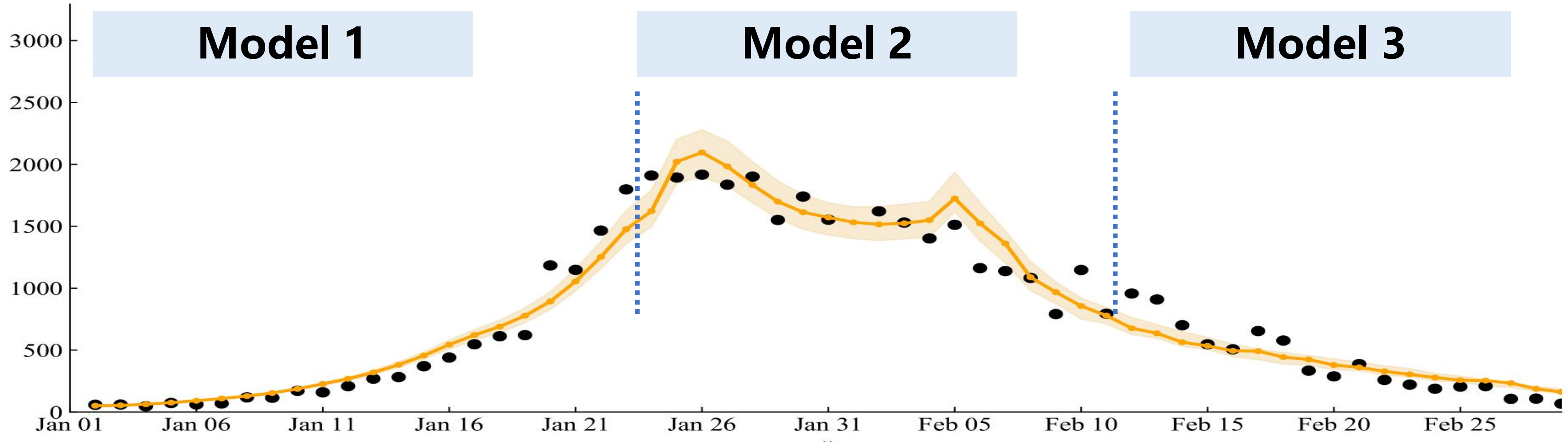






Base Model - SEIR-class (2)

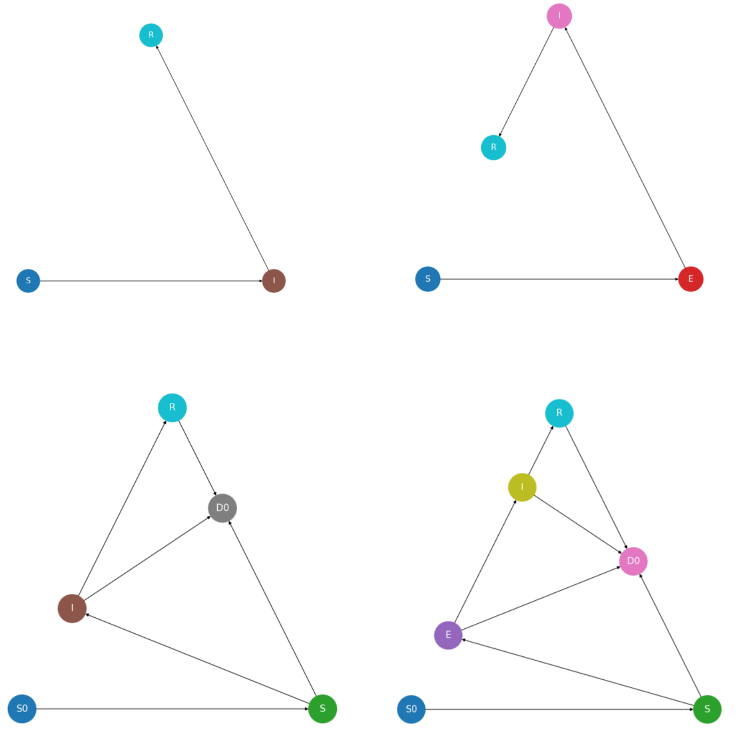
- Every models in LibEpidemic can use multi-stage or dynamic-parameters strategy
- Multi-stage means change the whole model in different time part



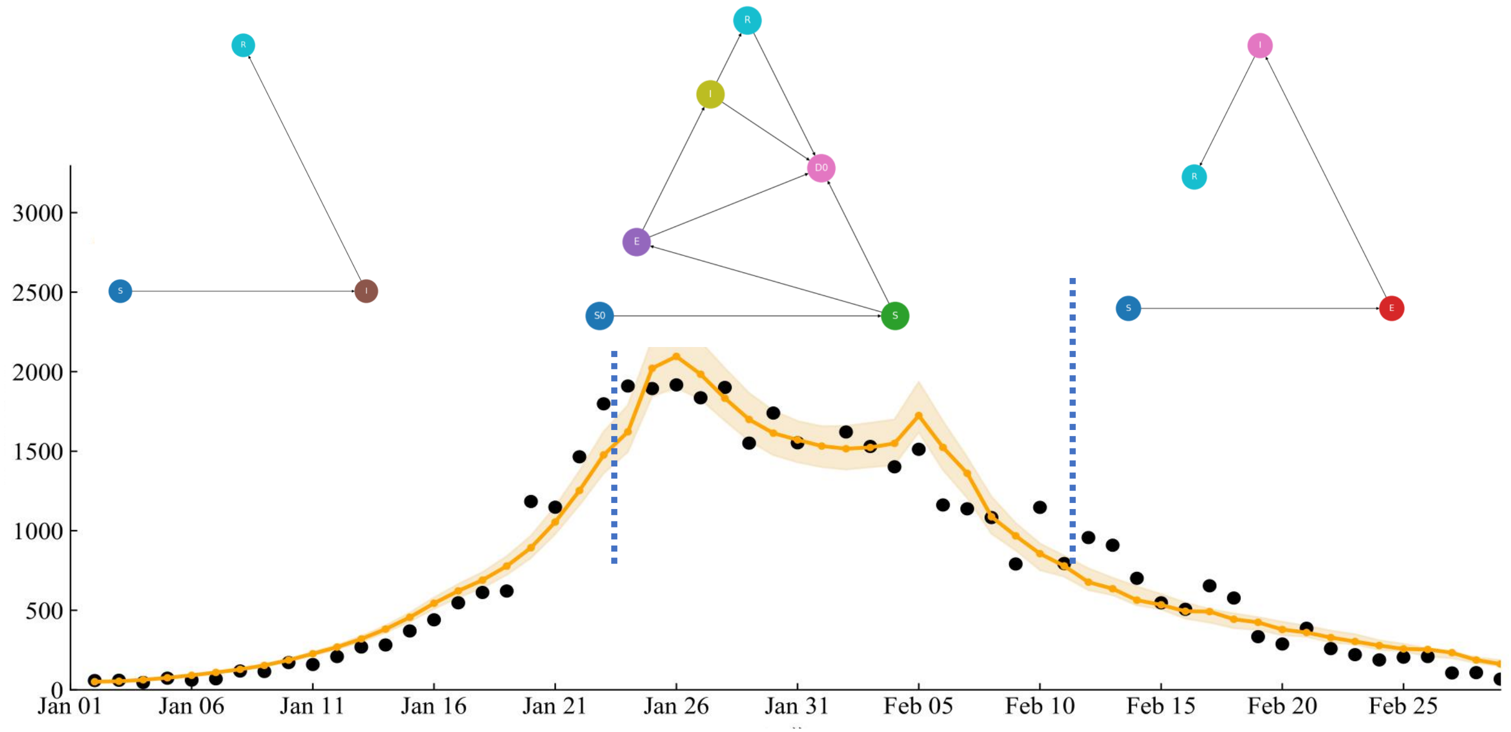
- Dynamic-parameters means only the parameters can change with time, not the structure

Base Model - SEIR-class (3)

- Considering that the inputs to the underlying SEIR model are similar, LibEpidemic provide Mixed-strategy Model to Epidemiologists . The mixed-strategy model is used with the multi-stage method, and **the optimal model in the model library is automatically selected** through the given loss function in each stage.



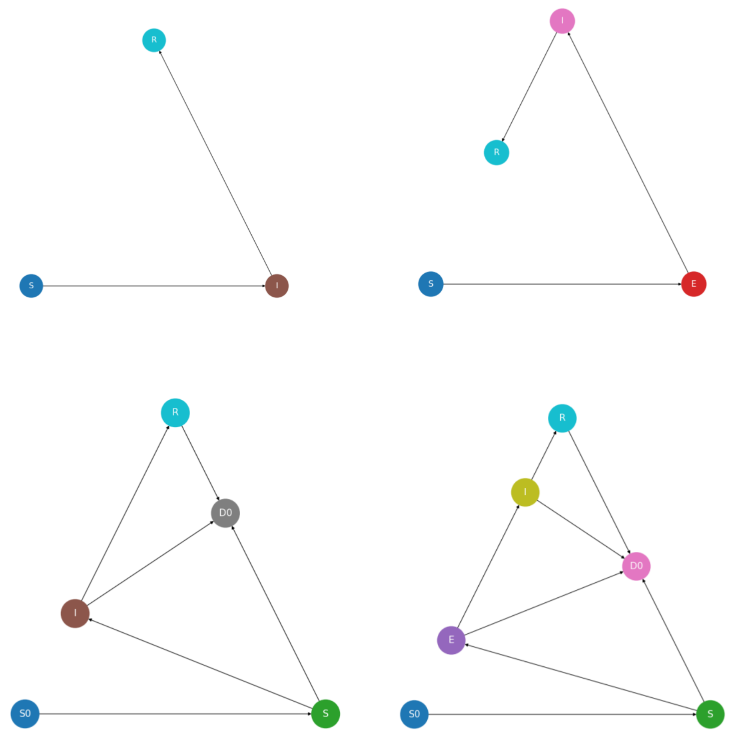
Model Library



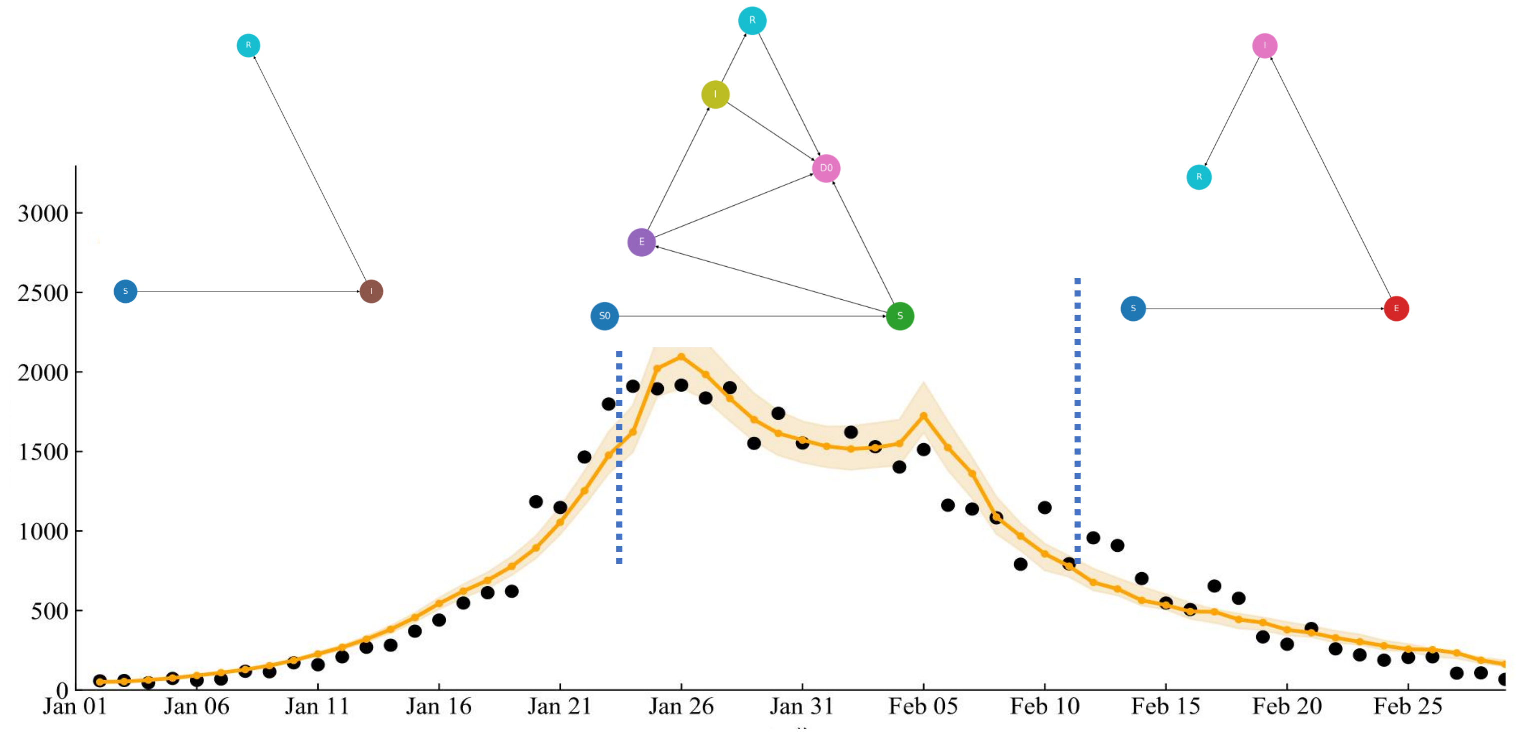
Select Result

Base Model - SEIR-class (3)

- Mixed-strategy models can be used to build a "one fits all model“. Using this model, combined with multi-stage global predictions (tests), the total error is less than or equal to any single model.



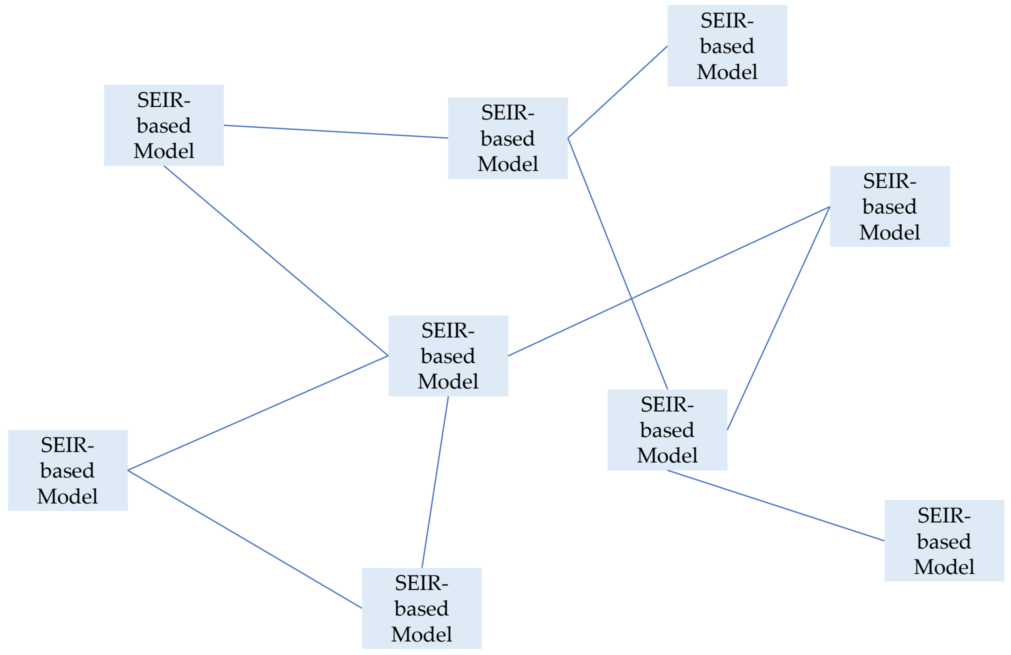
Model Library



Select Result

Base Model – Metapopulation Model

- In the SEIR model, ”transmission” occurs uniformly between susceptible and infected compartments. But actually, in human society, this process goes along social networks.
- The metapopulation model regards each node on the social network as a population, and the transmission of infectious diseases within the population is uniform, which is in line with the SEIR model. Between populations, along every edge on a social network, there is flow of individuals.
- LibEpidemic provide a complete example of building a metapopulation model



Applications – Complex Model from Paper

Applications – Complex Model from Paper (2)

Applications – Complex Model from Paper (3)

Applications – Complex Model from Paper (4)

Applications – Complex Model from Paper (5)

Applications - NPIs

Thanks for Listening!

王静远，北京航空航天大学

Jingyuan Wang, Beihang University, Beijing, China

jywang@buaa.edu.cn，http://www.bigcity.ai