Scripted Processes in NepidemiX Tutorial

Lukas Ahrenberg
(ahrenberg@irmacs.sfu.ca)

2014-03-30

Times

08:30 – 11:30	Lectures and tea/coffee
11:30 – 13:00	Computer room
13:00 – 13:45	Lunch and discussions
13:45 – 14:45	Computer room
15:00 – 16:00	Discussion and coffee

Goals

Theory

- Network modelling overview
- Introduction to NepidemiX

Practice

- Simulation of two testing strategies
- Uniform regular testing
- Motivated testing using social dynamics

Discussion

- Analysis of results
- Feedback

Motivation

HIV testing and treatment strategies.
Exploration of two (extreme) testing models. Uniform random testing versus testing motivation spreading as a 'meme'.

Model

Disease model

Susceptible ⟶ Acute ⟶ Chronic

Treatment

Infected nodes that are tested are put on immediate 100% effective treatment.

Uniform (regular) testing

- Same chance of getting tested for all

Motivated testing

- Majority of the population tested on average very seldom
- Motivated minority gets tested often
- Motivation spreads as an SIS process on the network

Model

Disease model

Susceptible ⟶ Acute ⟶ Chronic

Treatment

Infected nodes that are tested are put on immediate 100% effective treatment.

Uniform (regular) testing

- Same chance of getting tested for all

Motivated testing

- Majority of the population tested on average very seldom
- Motivated minority gets tested often
- Motivation spreads as an SIS process on the network

Model

Disease model

Susceptible ⟶ Acute ⟶ Chronic

Treatment

Infected nodes that are tested are put on immediate 100% effective treatment.

Uniform (regular) testing

- Same chance of getting tested for all

Motivated testing

- Majority of the population tested on average very seldom
- Motivated minority gets tested often
- Motivation spreads as an SIS process on the network

Model

Disease model

Susceptible ⟶ Acute ⟶ Chronic

Treatment

Infected nodes that are tested are put on immediate 100% effective treatment.

Uniform (regular) testing

- Same chance of getting tested for all

Motivated testing

- Majority of the population tested on average very seldom
- Motivated minority gets tested often
- Motivation spreads as an SIS process on the network

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Simulations overview

- Basic disease model on a scale-free network
- Investigate two testing strategies
- Uniform (random) testing
- Motivated testing

You will...

Write NepidemiX scripts for the two testing models, run them, and look at the incidence.

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for … for … if … elseif … elseif … elseif … elseif … elseif .. elseif …

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for ... for ... if ... elseif ... elseif ... elseif ... elseif ... elseif .. elseif ...

NepidemiX will

1. Read your configuration

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for … for … if … elseif … elseif … elseif … elseif … elseif .. elseif …

NepidemiX will

1. Read your configuration
2. Read your process description

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for ... for ... if ... elseif ... elseif ... elseif ... elseif ... elseif .. elseif ...

NepidemiX will

1. Read your configuration
2. Read your process description
3. Build a network

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for … for … if … elseif … elseif … elseif … elseif … elseif .. elseif …

NepidemiX will

1. Read your configuration
2. Read your process description
3. Build a network
4. Iterate a specific number of time steps

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for ... for ... if ... elseif ... elseif ... elseif ... elseif ... elseif .. elseif ...

NepidemiX will

1. Read your configuration
2. Read your process description
3. Build a network
4. Iterate a specific number of time steps
5. Execute process for every node in every step

NepidemiX overview

A large class of simulations have the same structure

- Contact topology in for of a network
- Attributes describe states
- A process how states change
- State changes are expressed as probability per time step
- for … for … if … elseif … elseif … elseif … elseif … elseif .. elseif …

NepidemiX will

1. Read your configuration
2. Read your process description
3. Build a network
4. Iterate a specific number of time steps
5. Execute process for every node in every step
6. Save loads of data

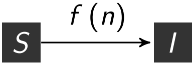
NepidemiX concepts

Processes usually take the form of a diagram describing statements

NepidemiX concepts

Processes usually take the form of a diagram describing statements

> *The probability for a susceptible individual to get infected grows with the number of infected neighbours in the network.*



$f(n)$

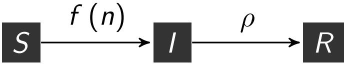
S

I

NepidemiX concepts

Processes usually take the form of a diagram describing statements

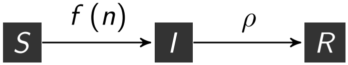
> *The probability for a susceptible individual to get infected grows with the number of infected neighbours in the network.*
>
> *The probability for an infected individual to recover is constant in unit time.*



NepidemiX concepts

Processes usually take the form of a diagram describing statements

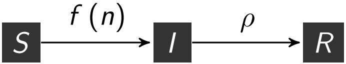
> *The probability for a susceptible individual to get infected grows with the number of infected neighbours in the network.*
>
> *The probability for an infected individual to recover is constant in unit time.*



NepidemiX concepts

Processes usually take the form of a diagram describing statements

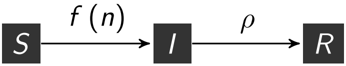
> *The probability for a susceptible individual to get infected grows with the number of infected neighbours in the network.*
>
> *The probability for an infected individual to recover is constant in unit time.*



NepidemiX concepts

Processes usually take the form of a diagram describing statements

> *The probability for a susceptible individual to get infected grows with the number of infected neighbours in the network.*
>
> *The probability for an infected individual to recover is constant in unit time.*



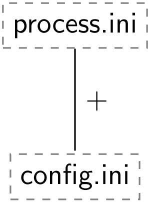
Discrete in time and space; Stochastic; Not compartmental.

High level use of NepidemiX

High level use of NepidemiX

process.ini

High level use of NepidemiX

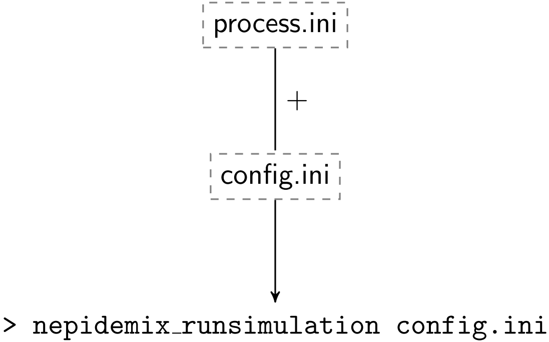


process.ini

+

config.ini

High level use of NepidemiX



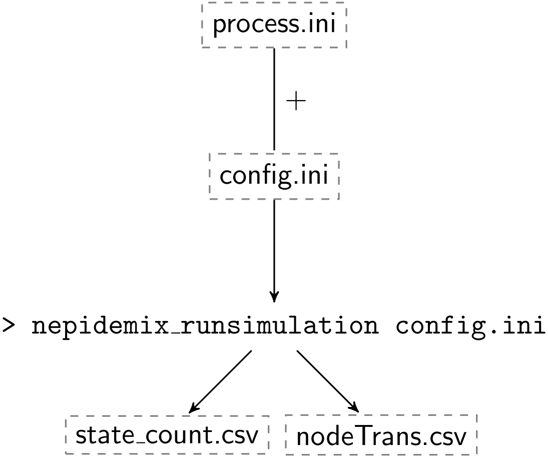
process.ini

+

config.ini

```
> nepidemix_runsimulation config.ini
```

High level use of NepidemiX



process.ini

+

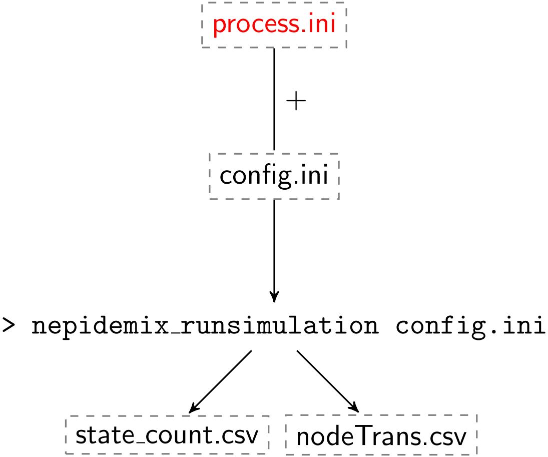
config.ini

```
> nepidemix_runsimulation config.ini
```

state_count.csv

nodeTrans.csv

High level use of NepidemiX



process.ini

+

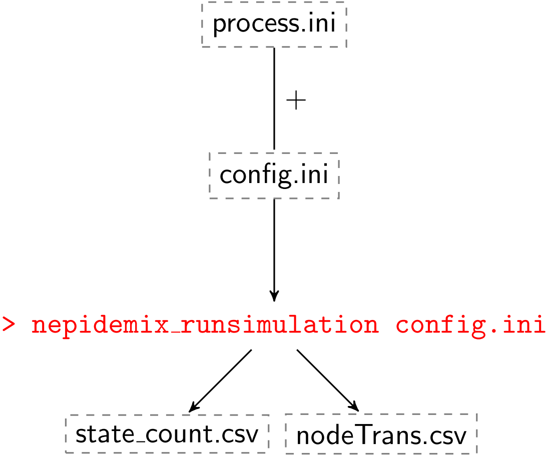
config.ini

```
> nepidemix_runsimulation config.ini
```

state_count.csv

nodeTrans.csv

High level use of NepidemiX



process.ini

+

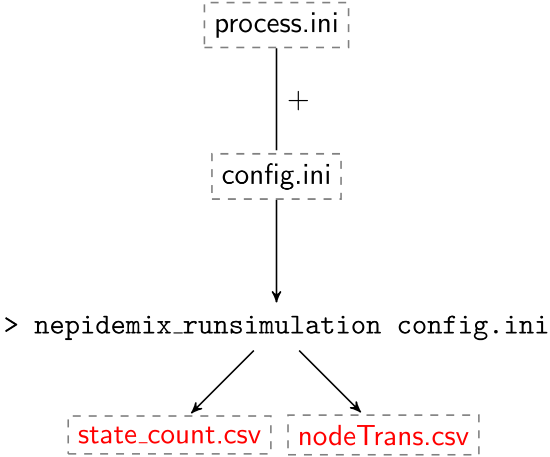
config.ini

```
> nepidemix_runsimulation config.ini
```

state_count.csv

nodeTrans.csv

High level use of NepidemiX



process.ini

+

config.ini

```
> nepidemix_runsimulation config.ini
```

state_count.csv

nodeTrans.csv

Three processes

1. Disease process
2. Testing process
3. Motivation spread model

We'll show the process and then how to script it to NepidemiX.

Disease model description

- Three disease states states

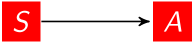
Disease model description

- Three disease states states
    - Susceptible (S)

S

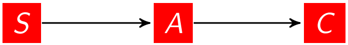
Disease model description

- Three disease states states
    - Susceptible (S)
    - Acute infection (A)



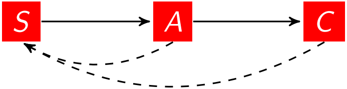
Disease model description

- Three disease states states
    - Susceptible (S)
    - Acute infection (A)
    - Chronic infection (C)

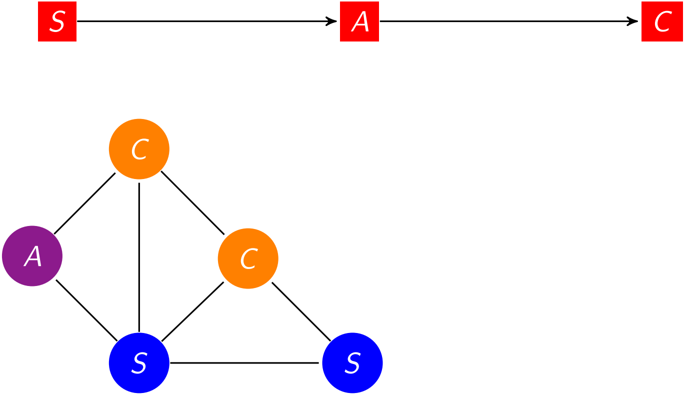


Disease model description

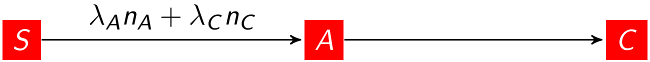
- Three disease states states
  - Susceptible (S)
  - Acute infection (A)
  - Chronic infection (C)
- Death is a transition to (S)

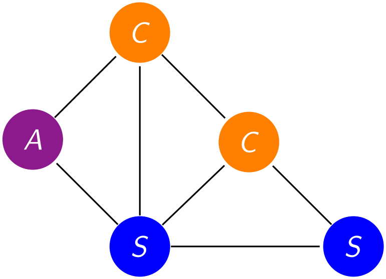


Disease transitions



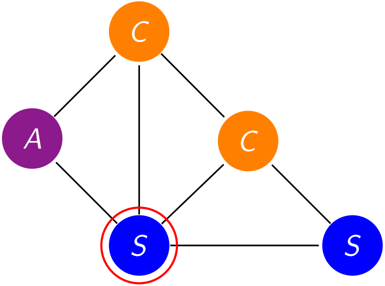
Disease transitions





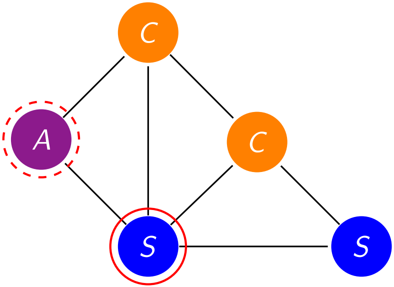
Disease transitions

$$\boxed{S} \xrightarrow{\lambda_A n_A + \lambda_C n_C} \boxed{A} \longrightarrow \boxed{C}$$



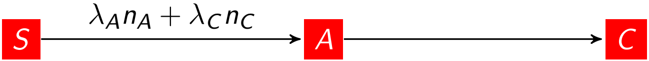
Disease transitions

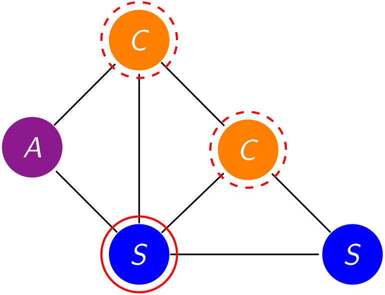
$$S \xrightarrow{\lambda_A n_A + \lambda_C n_C} A \longrightarrow C$$



$\lambda_A \times 1$

Disease transitions

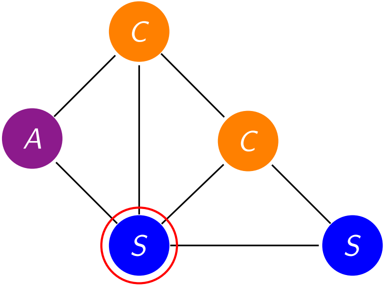




$$\lambda_A \times 1 + \lambda_C \times 2$$

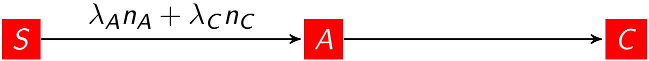
Disease transitions

$$S \xrightarrow{\lambda_A n_A + \lambda_C n_C} A \longrightarrow C$$

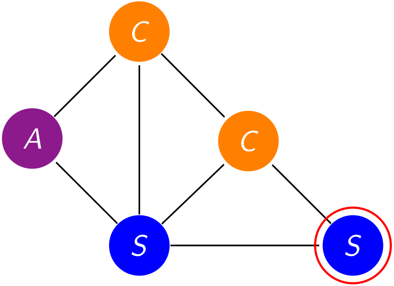


$$\lambda_A + 2\lambda_C$$

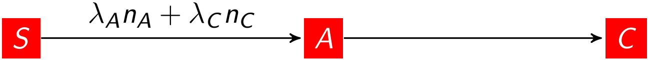
Disease transitions

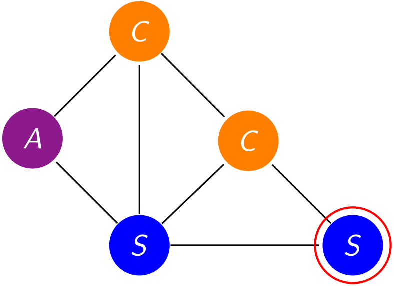


$$S \xrightarrow{\lambda_A n_A + \lambda_C n_C} A \longrightarrow C$$



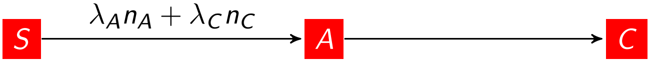
Disease transitions

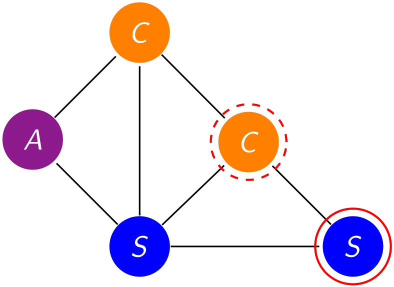




$\lambda_A \times 0$

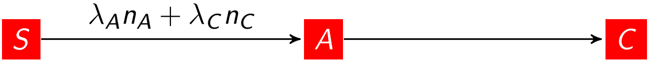
Disease transitions

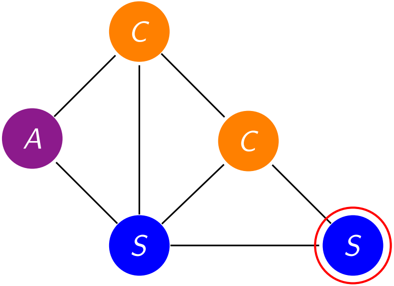




$$\lambda_A \times 0 + \lambda_C \times 1$$

Disease transitions

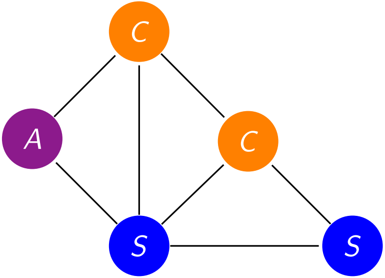




λ_C

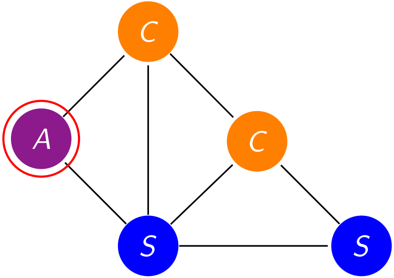
Disease transitions

$$S \xrightarrow{\lambda_A n_A + \lambda_C n_C} A \xrightarrow{\beta} C$$



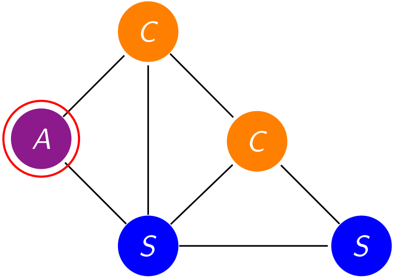
Disease transitions

$$S \xrightarrow{\lambda_A n_A + \lambda_C n_C} A \xrightarrow{\beta} C$$



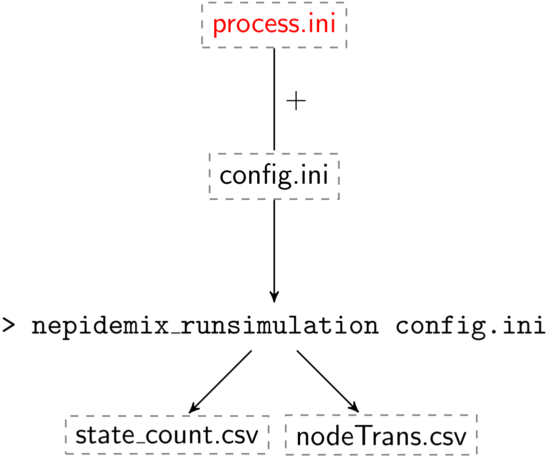
Disease transitions

$$S \xrightarrow{\lambda_A n_A + \lambda_C n_C} A \xrightarrow{\beta} C$$



β

Disease process in NepidemiX



process.ini

+

config.ini

```
> nepidemix_runsimulation config.ini
```

state_count.csv

nodeTrans.csv

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

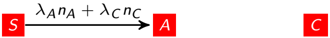
S A C

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

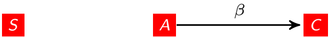
Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```



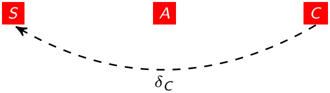
Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```



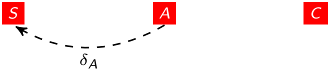
Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```



Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```



Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

Disease process in NepidemiX

```
[NodeAttributes]
# Susceptible (S), Acute (A), Chronic (C)
disease = S, A, C

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A *NN({disease :  A}) + lambda_C *NN({disease :  C})
{disease :  A} ->{disease :  C} = beta
# Death
{disease :  C} ->{disease :  S} = delta_C
{disease :  A} ->{disease :  S} = delta_A

[MeanFieldStates]
{}
```

Testing model description

- Two states

Testing model description

- Two states
- Untreated

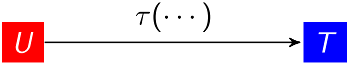
Testing model description

- Two states
- Untreated
- Treatment

U T

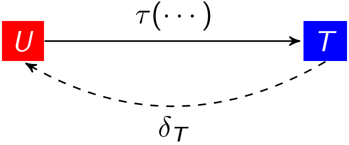
Testing model description

- Two states
- Untreated
- Treatment
- Probability of test depends on method
- Simplification: treatment is immediate

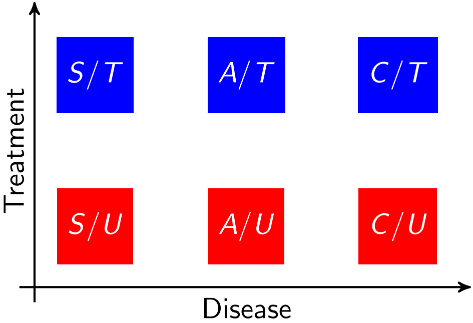


Testing model description

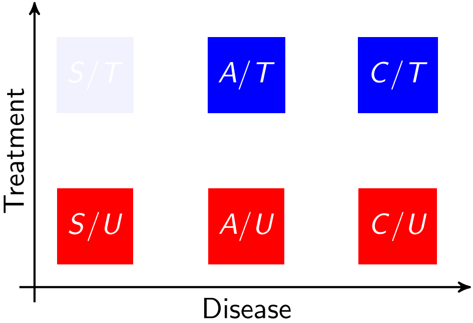
- Two states
- Untreated
- Treatment
- Probability of test depends on method
- Simplification: treatment is immediate
- Being treated will improve life expectancy



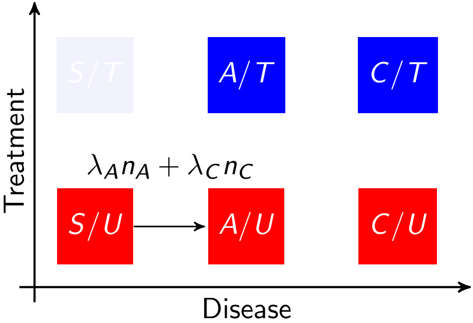
Disease model with testing and treatment



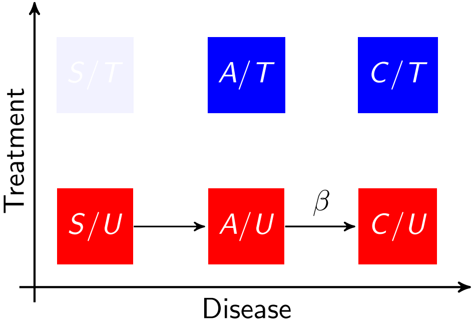
Disease model with testing and treatment



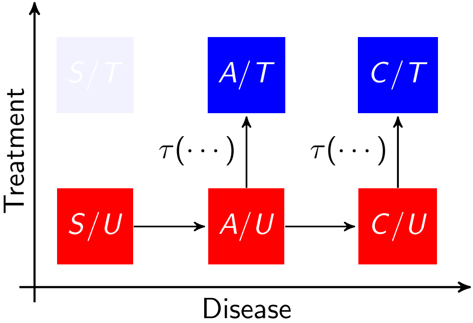
Disease model with testing and treatment



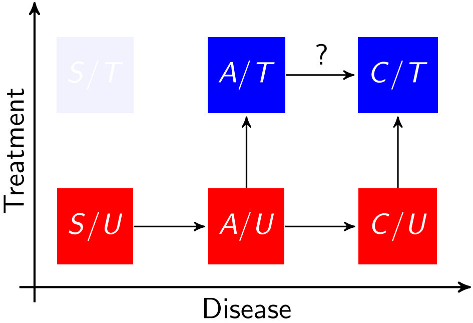
Disease model with testing and treatment



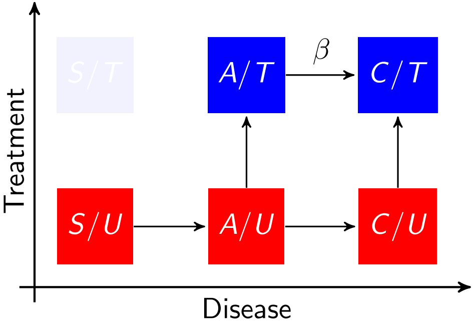
Disease model with testing and treatment



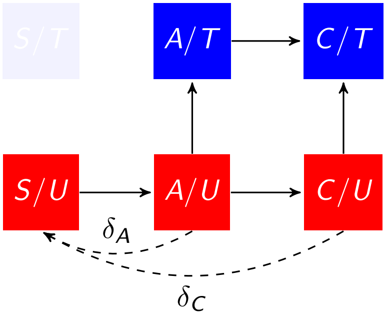
Disease model with testing and treatment



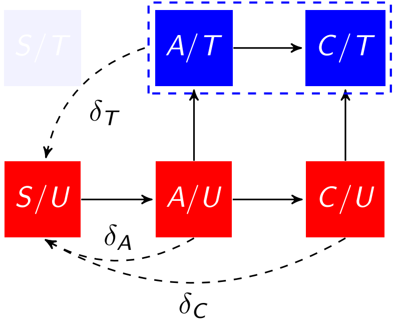
Disease model with testing and treatment



Disease model with testing and treatment



Disease model with testing and treatment

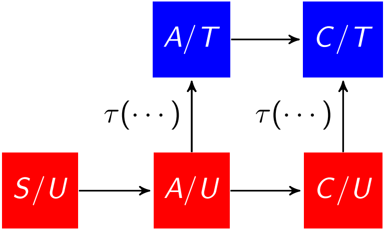


Uniform testing

Uniform regular testing - everyone gets tested equally often on average

Uniform, regular testing model

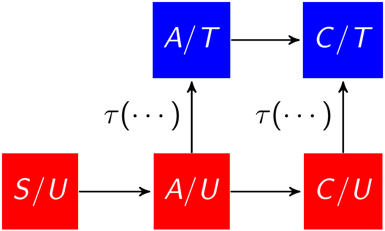
- Define $\tau(\cdots)$



$$\begin{tikzcd}
 & A/T \arrow[r] & C/T \\
S/U \arrow[r] & A/U \arrow[r] \arrow[u, "\tau(\cdots)"] & C/U \arrow[u, "\tau(\cdots)"]
\end{tikzcd}$$

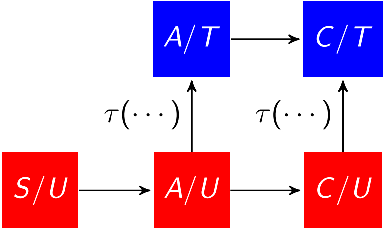
Uniform, regular testing model

- Define $\tau(\cdots)$
- Everyone gets tested on average the same number of times per life.



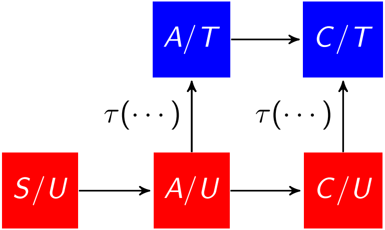
Uniform, regular testing model

- Define $\tau(\cdots)$
- Everyone gets tested on average the same number of times per life.
- Uniform and regular testing. Everyone has the same chance per unit time.



Uniform, regular testing model

- Define $\tau(\cdots)$
- Everyone gets tested on average the same number of times per life.
- Uniform and regular testing. Everyone has the same chance per unit time.
- Assume: chance of test is independent of disease stage.



A/T

C/T

$\tau(\cdots)$

$\tau(\cdots)$

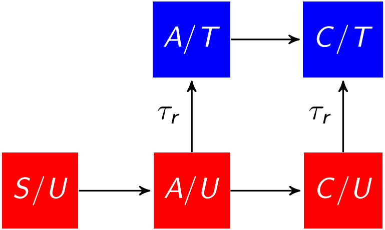
S/U

A/U

C/U

Uniform, regular testing model

- Define $\tau(\cdots)$
- Everyone gets tested on average the same number of times per life.
- Uniform and regular testing. Everyone has the same chance per unit time.
- Assume: chance of test is independent of disease stage.
- Constant rate, τ_r.



Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```

Random testing process in NepidemiX

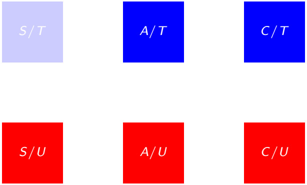
```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U} ->{treatment :  T} = tau_r

#Death
{treatment :  T} ->{disease :  S, treatment :  U} = delta_T
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U} = delta_A
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U} = delta_C

[MeanFieldStates]
{}
```

Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U} ->{treatment :  T} = tau_r

#Death
{treatment :  T} ->{disease :  S, treatment :  U} = delta_T
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U} = delta_A
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U} = delta_C

[MeanFieldStates]
{}
```

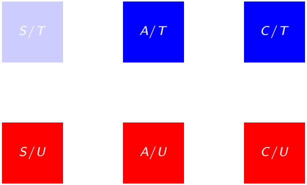
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```



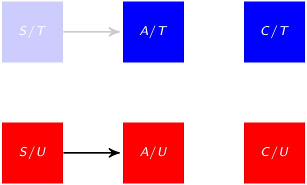
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U} ->{treatment :  T} = tau_r

#Death
{treatment :  T} ->{disease :  S, treatment :  U} = delta_T
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U} = delta_A
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U} = delta_C

[MeanFieldStates]
{}
```



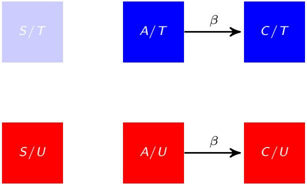
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```



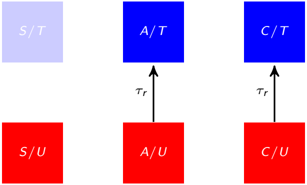
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```



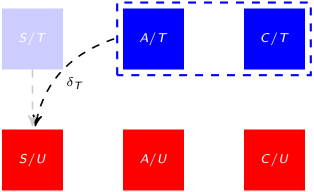
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U} ->{treatment :  T} = tau_r

#Death
{treatment :  T} ->{disease :  S, treatment :  U} = delta_T
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U} = delta_A
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U} = delta_C

[MeanFieldStates]
{}
```



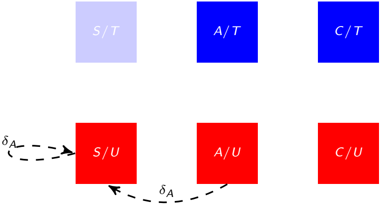
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U} ->{treatment :  T} = tau_r

#Death
{treatment :  T} ->{disease :  S, treatment :  U} = delta_T
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U} = delta_A
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U} = delta_C

[MeanFieldStates]
{}
```



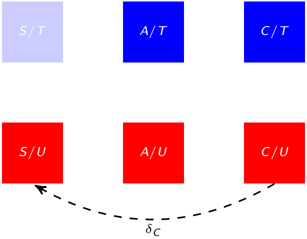
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```



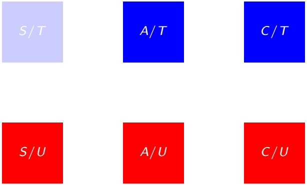
Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```



Random testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U} ->{treatment : T} = tau_r

#Death
{treatment : T} ->{disease : S, treatment : U} = delta_T
{disease : (S,A), treatment : U} ->{disease : S, treatment : U} = delta_A
{disease : C, treatment : U} ->{disease : S, treatment : U} = delta_C

[MeanFieldStates]
{}
```



Motivated testing

Motivated testing - motivation to get tested is caused by peer pressure

Motivated testing model

- Everyone has a small base chance of getting tested

Motivated testing model

- Everyone has a small base chance of getting tested
- A few are motivated

Motivated testing model

- Everyone has a small base chance of getting tested
- A few are motivated
- Motivation means much higher testing rate

Motivated testing model

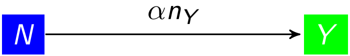
- Everyone has a small base chance of getting tested
- A few are motivated
- Motivation means much higher testing rate
- Motivated motivate their friends

Motivated testing model

- Everyone has a small base chance of getting tested
- A few are motivated
- Motivation means much higher testing rate
- Motivated motivate their friends
- Two states: No (unmotivated) and Yes (motivated)

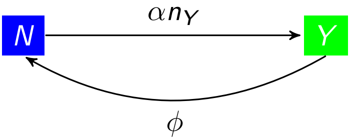
Motivated testing model

- Everyone has a small base chance of getting tested
- A few are motivated
- Motivation means much higher testing rate
- Motivated motivate their friends
- Two states: No (unmotivated) and Yes (motivated)
- Word of mouth

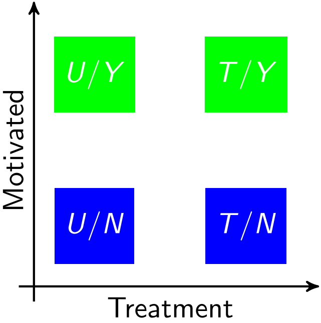


Motivated testing model

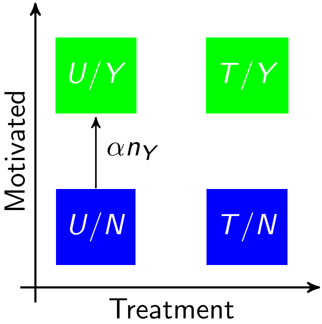
- Everyone has a small base chance of getting tested
- A few are motivated
- Motivation means much higher testing rate
- Motivated motivate their friends
- Two states: No (unmotivated) and Yes (motivated)
- Word of mouth
- Motivation is lost.



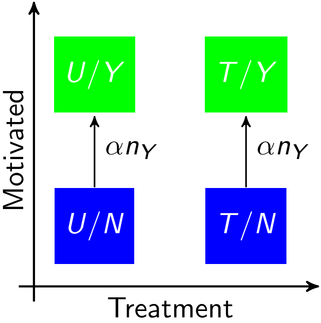
Motivation / Testing



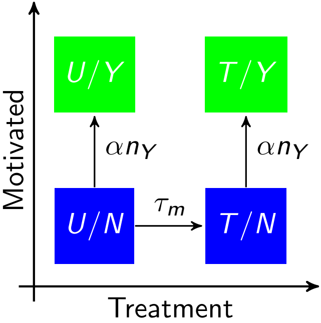
Motivation / Testing



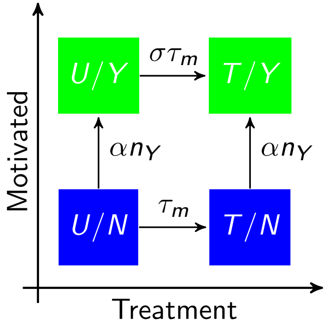
Motivation / Testing



Motivation / Testing



Motivation / Testing



U/Y

$\sigma\tau_m$

T/Y

αn_Y

αn_Y

U/N

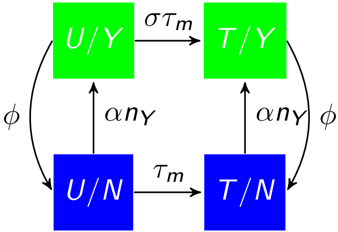
τ_m

T/N

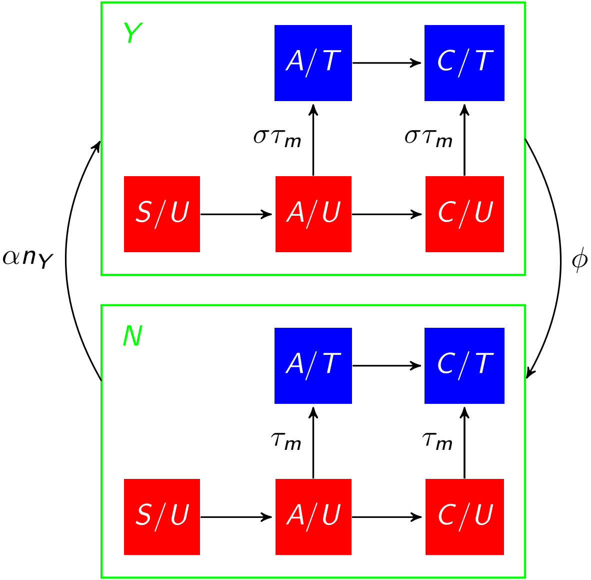
Motivated

Treatment

Motivation / Testing



Motivated testing model



Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```

Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```

Motivated testing process in NepidemiX

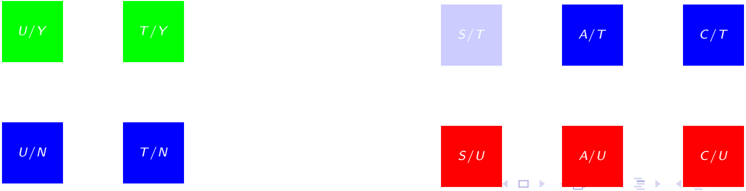
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```

Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```

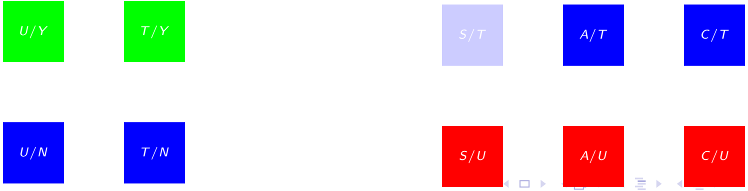
Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U, motivation : N} ->{treatment : T} = tau_m
{disease : (A,C), treatment : U, motivation : Y} ->{treatment : T} = tau_m * sigma

# Motivation to test
{motivation : N} ->{motivation : Y} = alpha * NN({motivation : Y})
{motivation :Y} ->{motivation : N} = phi

# Death
{treatment : T} ->{disease : S, treatment : U, motivation : N} = delta_T
{disease : C, treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_C
{disease : (S,A), treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_A

[MeanFieldStates]
{}
```



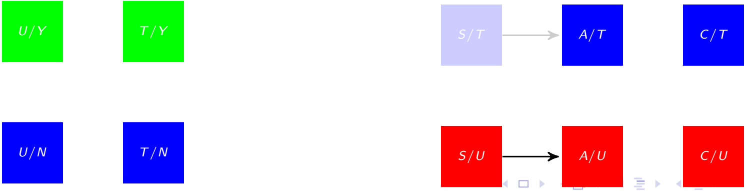
Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```



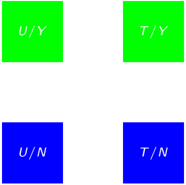
Motivated testing process in NepidemiX

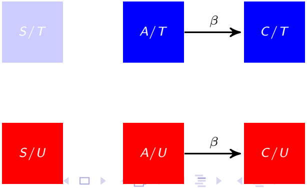
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```



Motivated testing process in NepidemiX

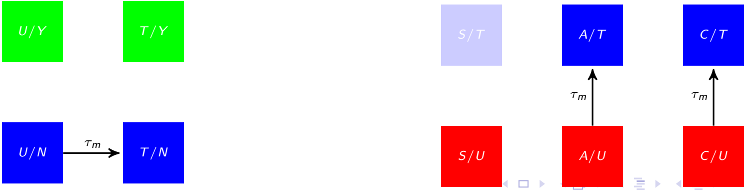
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U, motivation : N} ->{treatment : T} = tau_m
{disease : (A,C), treatment : U, motivation : Y} ->{treatment : T} = tau_m * sigma

# Motivation to test
{motivation : N} ->{motivation : Y} = alpha * NN({motivation : Y})
{motivation :Y} ->{motivation : N} = phi

# Death
{treatment : T} ->{disease : S, treatment : U, motivation : N} = delta_T
{disease : C, treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_C
{disease : (S,A), treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_A

[MeanFieldStates]
{}
```





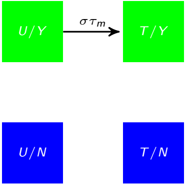
Motivated testing process in NepidemiX

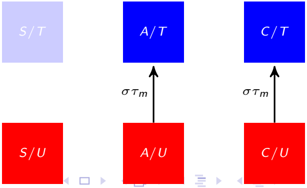
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```



Motivated testing process in NepidemiX

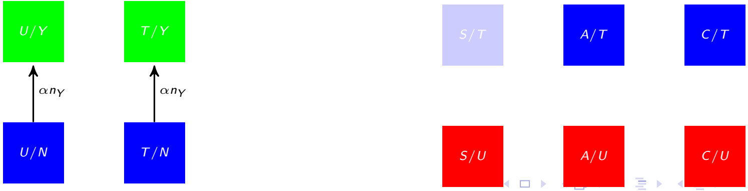
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U, motivation : N} ->{treatment : T} = tau_m
{disease : (A,C), treatment : U, motivation : Y} ->{treatment : T} = tau_m * sigma

# Motivation to test
{motivation : N} ->{motivation : Y} = alpha * NN({motivation : Y})
{motivation :Y} ->{motivation : N} = phi

# Death
{treatment : T} ->{disease : S, treatment : U, motivation : N} = delta_T
{disease : C, treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_C
{disease : (S,A), treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_A

[MeanFieldStates]
{}
```





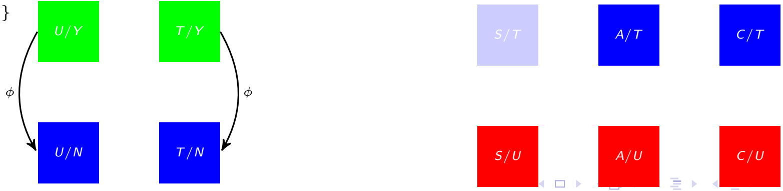
Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```



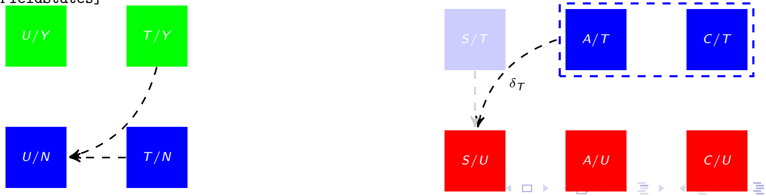
Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U, motivation : N} ->{treatment : T} = tau_m
{disease : (A,C), treatment : U, motivation : Y} ->{treatment : T} = tau_m * sigma

# Motivation to test
{motivation : N} ->{motivation : Y} = alpha * NN({motivation : Y})
{motivation :Y} ->{motivation : N} = phi

# Death
{treatment : T} ->{disease : S, treatment : U, motivation : N} = delta_T
{disease : C, treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_C
{disease : (S,A), treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_A

[MeanFieldStates]
{}
```



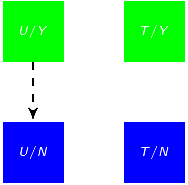
Motivated testing process in NepidemiX

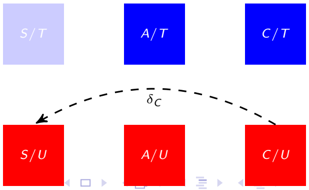
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U, motivation : N} ->{treatment : T} = tau_m
{disease : (A,C), treatment : U, motivation : Y} ->{treatment : T} = tau_m * sigma

# Motivation to test
{motivation : N} ->{motivation : Y} = alpha * NN({motivation : Y})
{motivation :Y} ->{motivation : N} = phi

# Death
{treatment : T} ->{disease : S, treatment : U, motivation : N} = delta_T
{disease : C, treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_C
{disease : (S,A), treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_A

[MeanFieldStates]
{}
```



Motivated testing process in NepidemiX

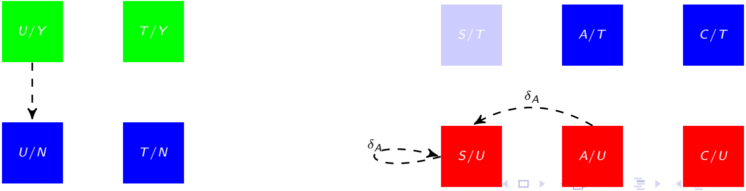
```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease :  S} ->{disease :  A} = lambda_A * NN({disease :  A, treatment :  U}) + lambda_C *
NN({disease :  C, treatment :  U})
{disease :  A} ->{disease :  C} = beta

# Testing and treatment
{disease :  (A,C), treatment :  U, motivation :  N} ->{treatment :  T} = tau_m
{disease :  (A,C), treatment :  U, motivation :  Y} ->{treatment :  T} = tau_m * sigma

# Motivation to test
{motivation :  N} ->{motivation :  Y} = alpha * NN({motivation :  Y})
{motivation :Y} ->{motivation :  N} = phi

# Death
{treatment :  T} ->{disease :  S, treatment :  U, motivation :  N} = delta_T
{disease :  C, treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_C
{disease :  (S,A), treatment :  U} ->{disease :  S, treatment :  U, motivation :  N} = delta_A

[MeanFieldStates]
{}
```





Motivated testing process in NepidemiX

```
[NodeAttributes]
disease = S, A, C
treatment = U, T
motivation = N, Y

[NodeRules]
# Infection
{disease : S} ->{disease : A} = lambda_A * NN({disease : A, treatment : U}) + lambda_C *
NN({disease : C, treatment : U})
{disease : A} ->{disease : C} = beta

# Testing and treatment
{disease : (A,C), treatment : U, motivation : N} ->{treatment : T} = tau_m
{disease : (A,C), treatment : U, motivation : Y} ->{treatment : T} = tau_m * sigma

# Motivation to test
{motivation : N} ->{motivation : Y} = alpha * NN({motivation : Y})
{motivation :Y} ->{motivation : N} = phi

# Death
{treatment : T} ->{disease : S, treatment : U, motivation : N} = delta_T
{disease : C, treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_C
{disease : (S,A), treatment : U} ->{disease : S, treatment : U, motivation : N} = delta_A

[MeanFieldStates]
{}
```



Experiment details

Disease and simulation parameters

Parameter	Value	Description
λ_A	0.64	Infectivity rate per acute contact
λ_C	0.016	Infectivity rate per chronic contact
β	6.0	Rate of 12/2 per year
δ_A	0.018	55 years (otherwise) healthy life
δ_C	0.1	10 years chronic life
dt	0.01	3.65 days

Experiment purpose

Compare uniform random testing with motivated testing.

Setting rates equal to compare testing

$$\tau_r = \tau_m \left(\langle \rho \rangle \sigma + (1 - \langle \rho \rangle) \right)$$

τ_r - Random testing rate, τ_m - Base (Un-) motivated testing rate.
σ - Motivation boost, $\bar{\rho}$ - fraction of motivated in population.

Testing parameters

Parameter	Value	Description
α	0.1	Testing meme inf.
ϕ	0.6	Testing meme removal

Testing parameters

Parameter	Value	Description
α	0.1	Testing meme inf.
ϕ	0.6	Testing meme removal

- Leads to 17% of the population being motivated on average

Testing parameters

Parameter	Value	Description
α	0.1	Testing meme inf.
ϕ	0.6	Testing meme removal

- Leads to 17% of the population being motivated on average

Uniform testing $\tau_r = 0.1$; One uniform random test per 10 years

Testing parameters

Parameter	Value	Description
α	0.1	Testing meme inf.
ϕ	0.6	Testing meme removal

- Leads to 17% of the population being motivated on average

Uniform testing $\tau_r = 0.1$; One uniform random test per 10 years

Motivated base chance $\tau_m = 0.02$; One test per 50 years

Motivation booster $\sigma = 25$; Motivated test every two years

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

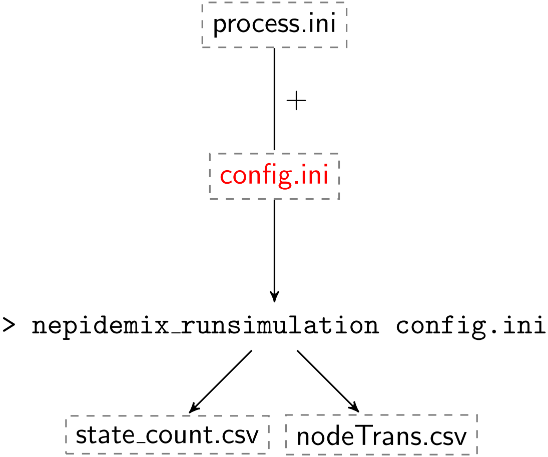
Question

Same testing rate, different strategies

Test 17% of population preferentially every two years and the remaining 83% once every fifty years, *or* test everyone every ten years?

Configuring nepidemix

- Parameter values
- Simulation settings
- Output settings



process.ini

+

config.ini

```
> nepidemix_runsimulation config.ini
```

state_count.csv

nodeTrans.csv

Random testing simulation config in NepidemiX

```
[Simulation]
iterations = 2000
dt = 0.01
process_class = ScriptedProcess
network_func = load_network
node_init = off
edge_init = off

[NetworkParameters]
file = network/disease_network.gpickle.bz2

[ProcessParameters]
file = uniform_process.ini
beta = 6.0
lambda_A = 0.64
lambda_C = 0.016
tau_r = 0.1
delta_A = 0.018
delta_C = 0.1
delta_T = delta_A

[Output]
output_dir = ./
base_name = uniform_testing
unique = no
save_state_count = yes
save_state_count_interval = 1
save_config = no
save_state_transition_cnt = yes
save_state_influx = no
```

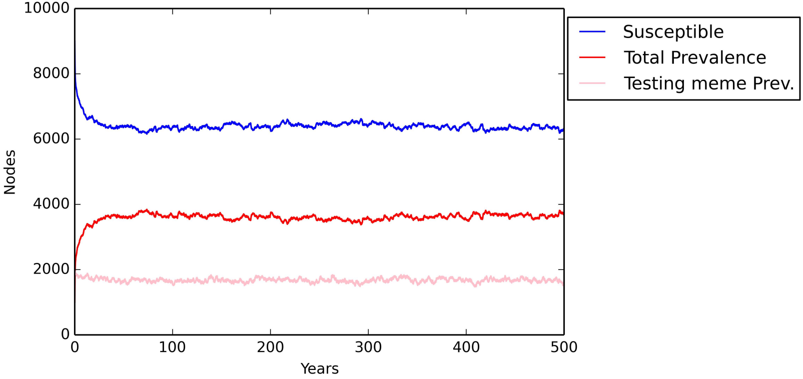
Motivated testing simulation config in NepidemiX

```
[Simulation]
iterations = 2000
dt = 0.01
process_class = ScriptedProcess
network_func = load_network
node_init = off
edge_init = off

[NetworkParameters]
file = network/disease_network.gpickle.bz2

[ProcessParameters]
file = motivated_process.ini
beta = 6
lambda_A = 0.64
lambda_C = 0.016
tau_m = 0.02
sigma = 25.0
alpha = 0.1
phi = 0.6
delta_A = 0.018
delta_C = 0.1
delta_T = delta_A

[Output]
output_dir = ./
base_name = motivated_testing
unique = no
save_state_count = yes
save_state_count_interval = 1
save_config = no
save_state_transition_cnt = yes
save_state_influx = no
```

The Network

- We are using a pre-prepared network
- Scale free, preferential attachement
- $\langle k \rangle \approx 6$
- Almost no clustering
- 10000 nodes
- Running disease and motivation processes for 10000 iterations
- No testing
- You will load it in your simulations

Disease process - Burn in time



Running the experiment

- Command line
- `> nepidemix_runsimulation config.ini`
- Yields csv files.

Output

- The node count
- The transition matrix
- For the tutorial: Extract incidence using ready script

Off to the computer room

Credits

by alegri / 4freephotos.com / wikimedia.org

by chichacha / flickr.com

by david 23 (David Thompson) / flickr.com