Finite resources, choices & infectious disease models

William Waites

Laboratory for Foundations of Computer Science
School of Informatics, University of Edinburgh

ICMS V-KEMS – Mathematical modelling and COVID-19
October 6th, 2020

- How can we represent changing behaviour in epi models?
- How can we represent finite resources in epi models?
- I mean *explicitly* represent...

- How can we represent changing behaviour in epi models?
- How can we represent finite resources in epi models?
- I mean *explicitly* represent...

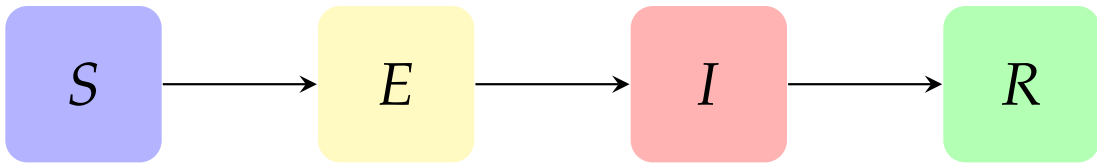
- How can we represent changing behaviour in epi models?
- How can we represent finite resources in epi models?
- I mean *explicitly* represent...

- Modelling entities: more than just people
- Focus on *interactions*

Rule-based modelling

- Used in molecular biology
- Generalisation of reaction-based models
- Transparent: explicit sytax for writing models
- Expressive: captures a large class of interesting models
- Scalable: keeps a lid on combinatorial explosion
- Composable: models can be easily combined

Compartmental models



Reaction models

$$S + I \xrightarrow{k_1} E + I \tag{1}$$

$$E \xrightarrow{k_2} I \tag{2}$$

$$I \xrightarrow{k_3} R \tag{3}$$

An epidemic of masks

$$P_{M-} + P_{M+} \longrightarrow P_{M+} + P_{M+} \tag{4}$$

$$P_{M+} + R \longrightarrow P_{M-} + R \tag{5}$$

Rule-based models

Agents

By analogy with *reagents*, RBM has *agents*:

```
// An person has a state describing the disease as
// well as receptors for vaccines and messages
%agent: P(covid{s e i r} vax msg)
// A vaccine can bond to a person
%agent: Vax(p)
// A message can be positive or negative
%agent: Msg(s{pos neg})
```

Rule-based models

Agents

```
// An person has a state describing the disease as
// well as receptors for vaccines and messages
%agent: P(covid{s e i r} vax msg)


P()  // any person
```

Rule-based models

Agents

```
// An person has a state describing the disease as
// well as receptors for vaccines and messages
%agent: P(covid{s e i r} vax msg)


P(covid{i})  // an infectious person
```

Rule-based models

Agents

```
// An person has a state describing the disease as
// well as receptors for vaccines and messages
%agent: P(covid{s e i r} vax msg)


P(vax[.])  // an unvaccinated person
P(vax[_])  // a vaccinated person
```

Rule-based models

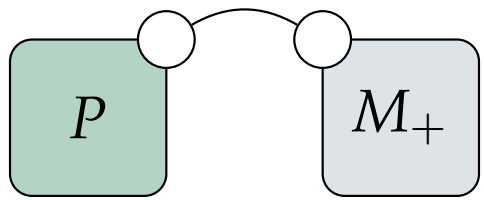
Agents

```
// An person has a state describing the disease as
// well as receptors for vaccines and messages
%agent: P(covid{s e i r} vax msg)


// an unvaccinated, susceptible person
P(covid{s}, vax[.])
```

Rule-based models

Agents

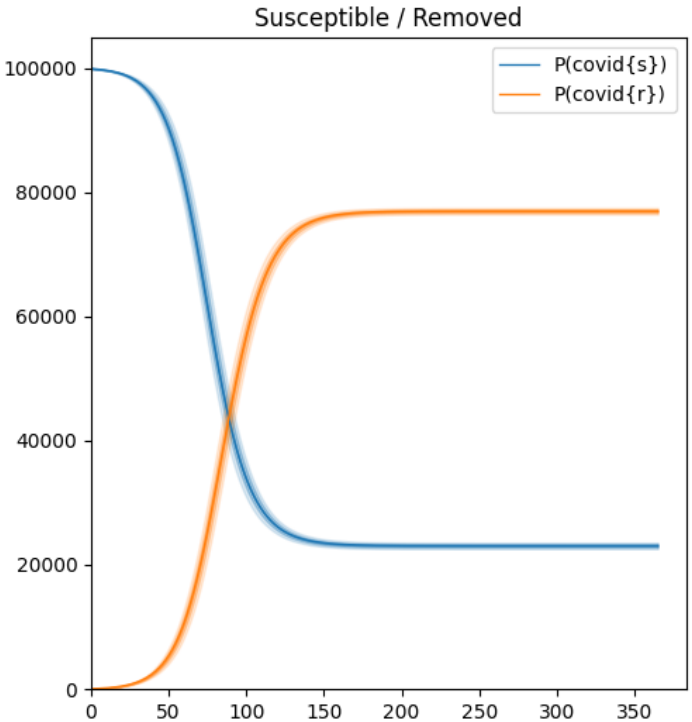
```
// An person has a state describing the disease as
// well as receptors for vaccines and messages
%agent: P(covid{s e i r} vax msg)
// A message can be positive or negative
%agent: Msg(s{pos neg})


// an with a positive opinion about vaccines
P(msg[1]), Msg(s[1]{p})
```

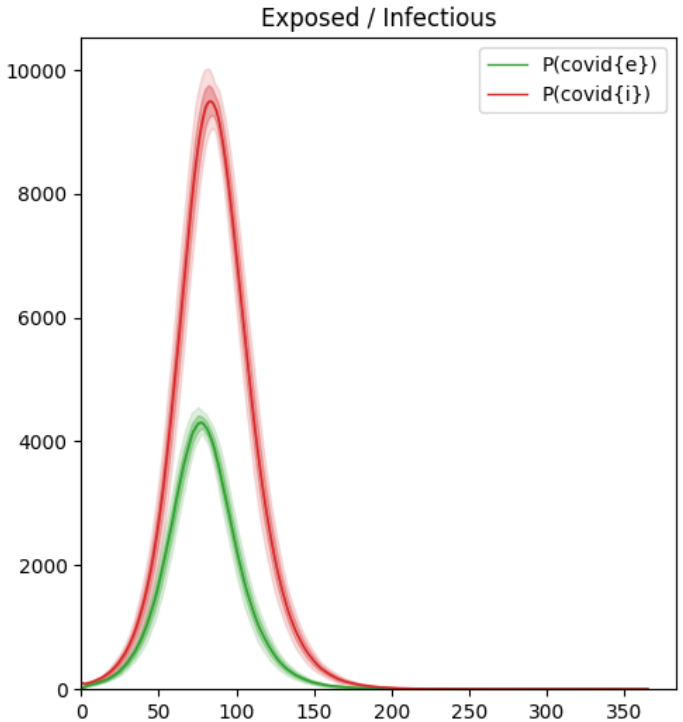


Rules: infection dynamics

```
// The usual SEIR dynamics. Caveat: only
// unvaccinated individuals become infected
'infection'     P(covid{s}, vax[.]), P(covid{i}) ->
                P(covid{e}, vax[.]), P(covid{i}) @ c*
                    beta/N
'progression'   P(covid{e}) -> P(covid{i}) @ alpha
'removal'       P(covid{i}) -> P(covid{r}) @ gamma
```

Baseline: just SEIR



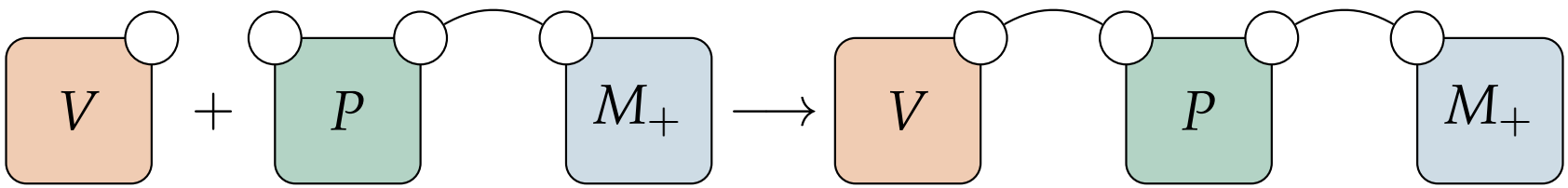


Rules: production and degradation of vaccine

```
// A factory producing vaccines
'factory'       . -> Vax() @ m
// After some time, the vaccine degrades
'degradation'  Vax() -> . @ deg
```

Rules: immunisation

```
// People with positve sentiment about vaccines
// get immunised -- bound to a vaccine
'immunisation'
    Vax(p[.]), P(vax[.], msg[1]), Msg(s[1]{pos}) ->
    Vax(p[2]), P(vax[2], msg[1]), Msg(s[1]{pos}) @ vax
```



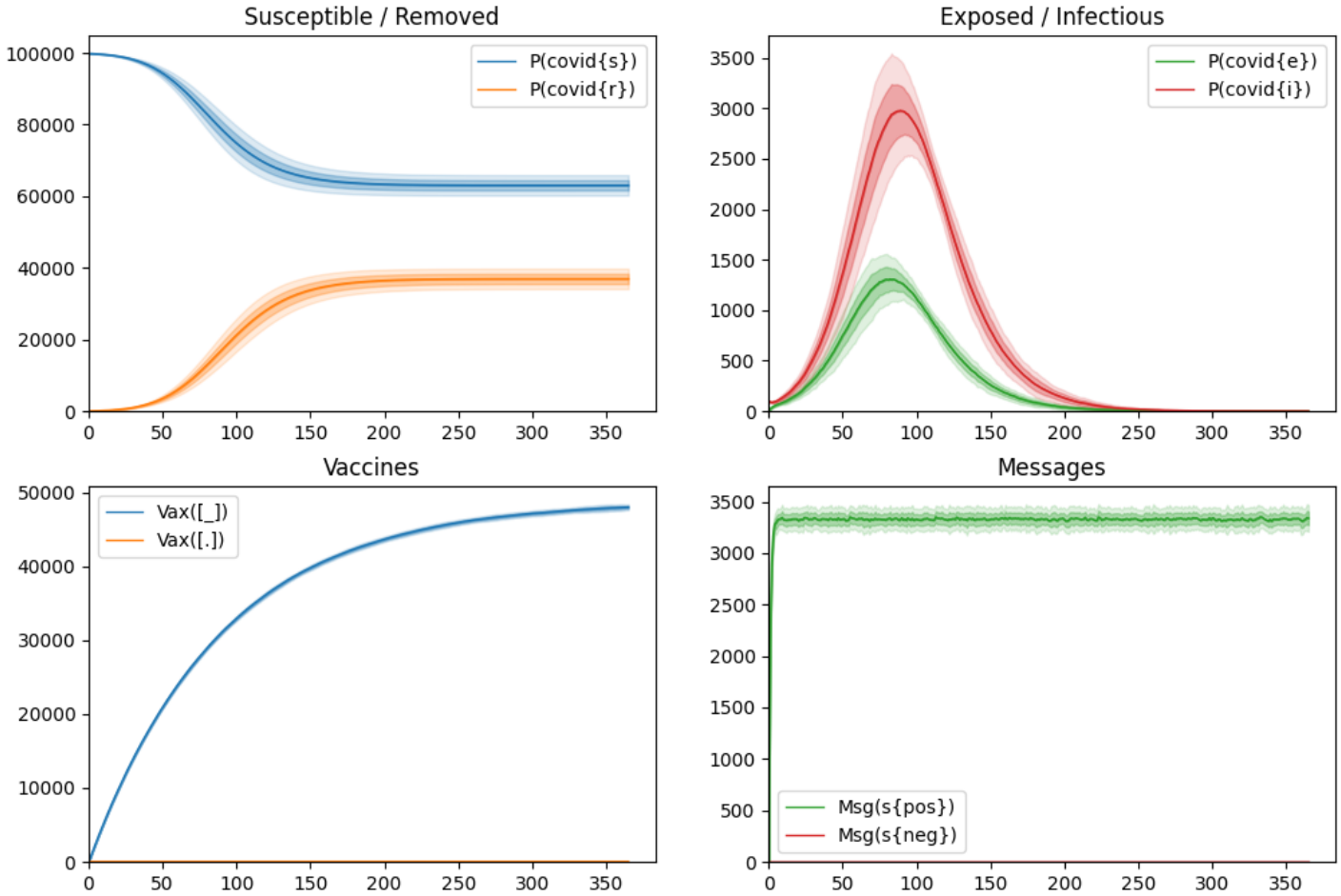
Rules: influence and forgetting

```
// People are influenced by messages, they become
// bound to a message
'influence'     P(msg[.]), Msg(s[.]) ->
                P(msg[1]), Msg(s[1]) @ hear

// Messages degrade at different rates depending
// if they are positive or negative
'forget_pos'    Msg(s{pos}) -> . @ dpos
'forget_neg'    Msg(s{neg}) -> . @ dneg
```

Rules: promotional campaign

```
// Some global promotion of vaccines -- perhaps
// an advertising campaign
'promotion'     . -> Msg(s{pos}) @ promo
```

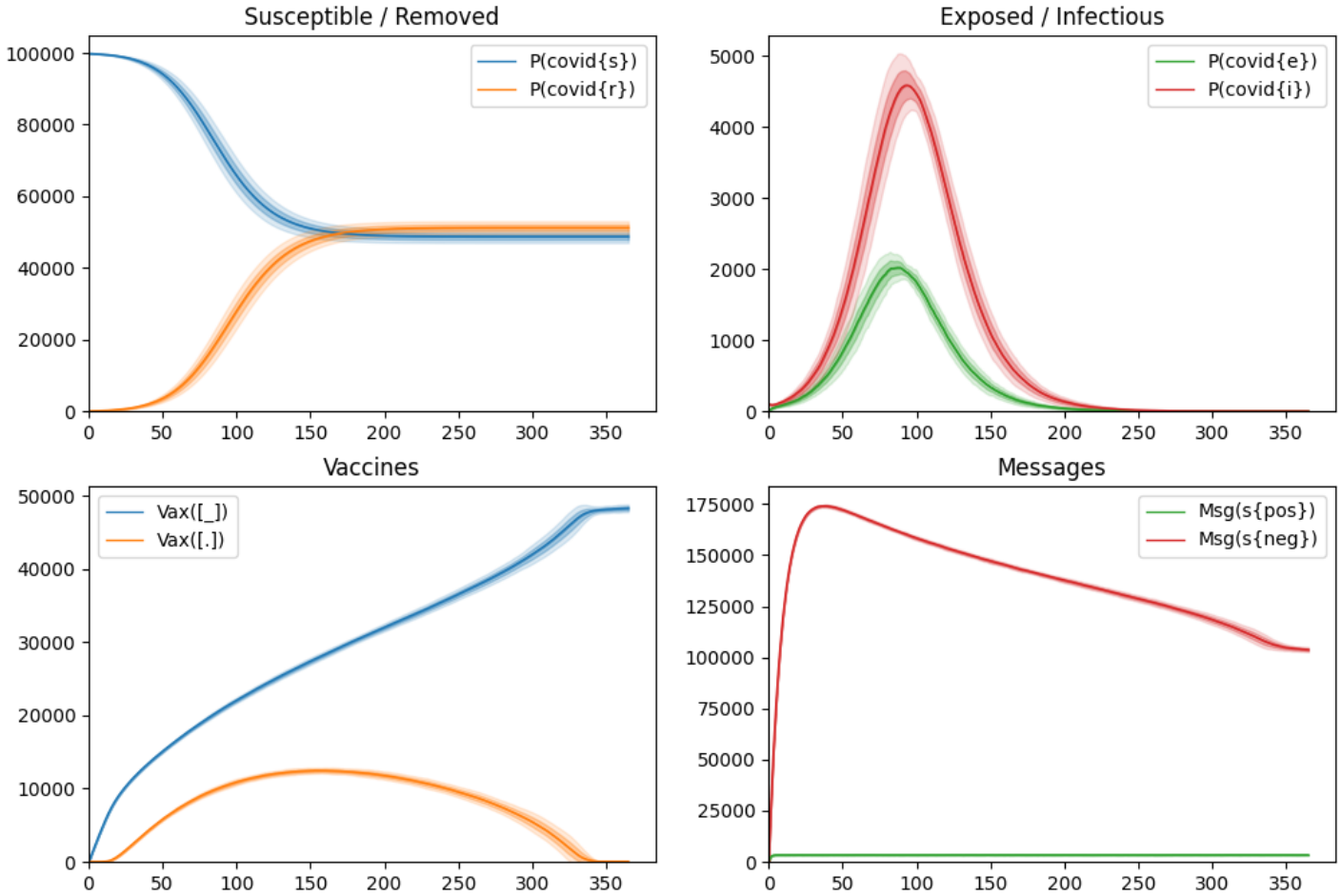
Promotional campaign



Rules: antivaxxers

```
// Negative sentiment is spread by some
// unvaccinated people.
'nonsense' P(vax[.]), . ->
           P(vax[.]), Msg(s{neg}) @ neg
```

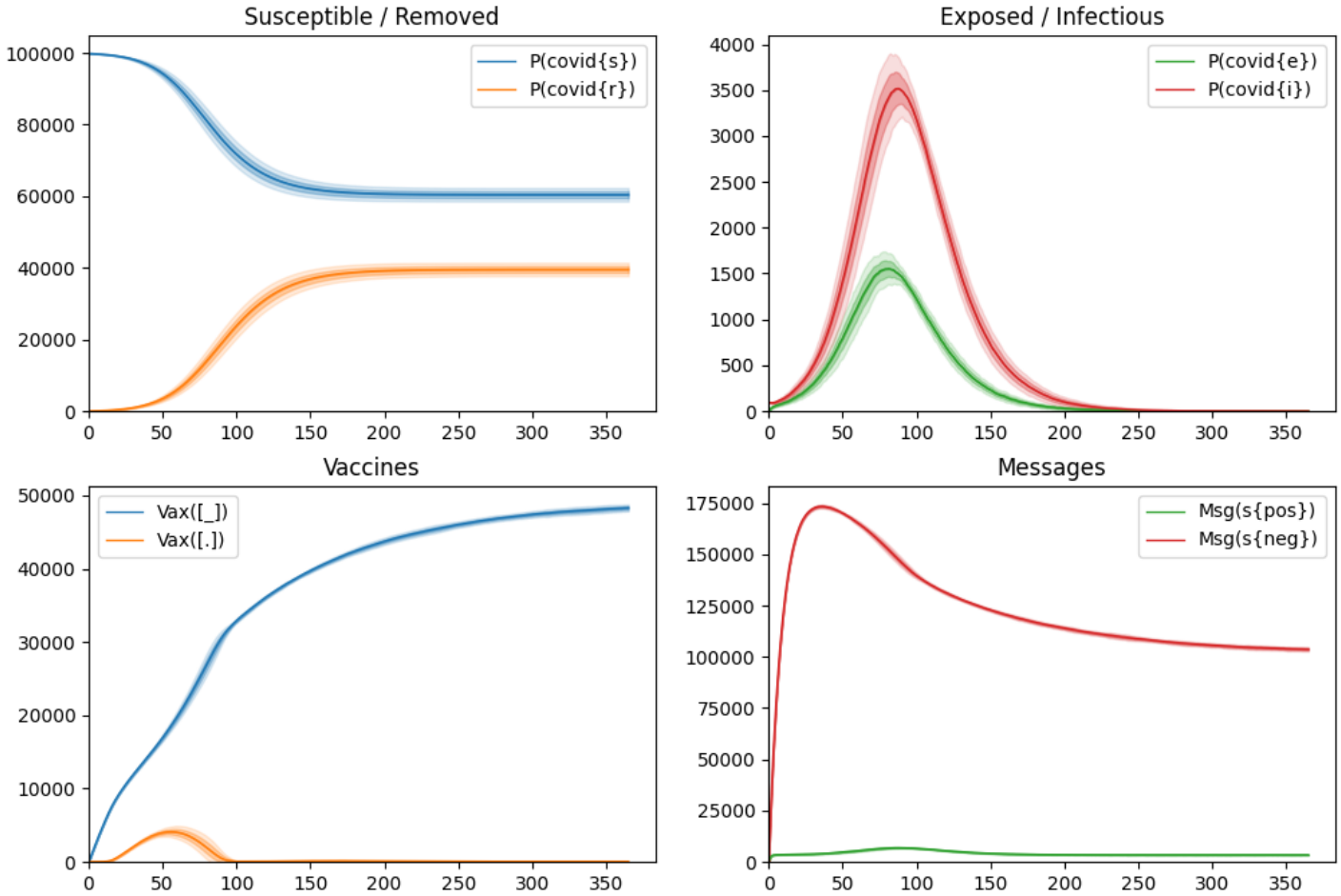
Antivaxxers



Rules: the voice of experience

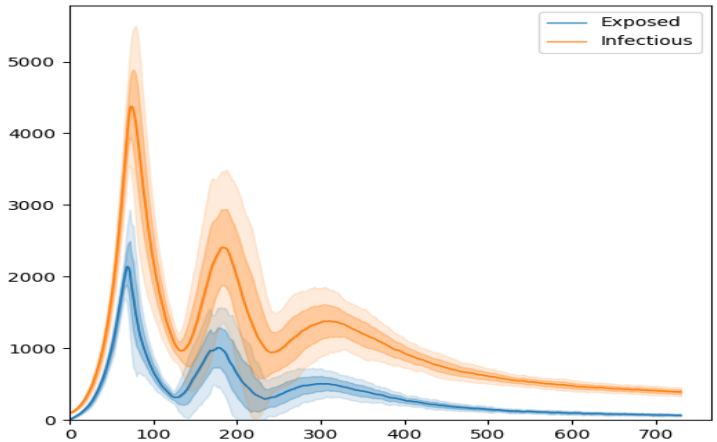
```
// Pro-vaccine messages proportional to infection
'experience' P(covid{i}), . ->
             P(covid{i}), Msg(s{pos}) @ exp
```

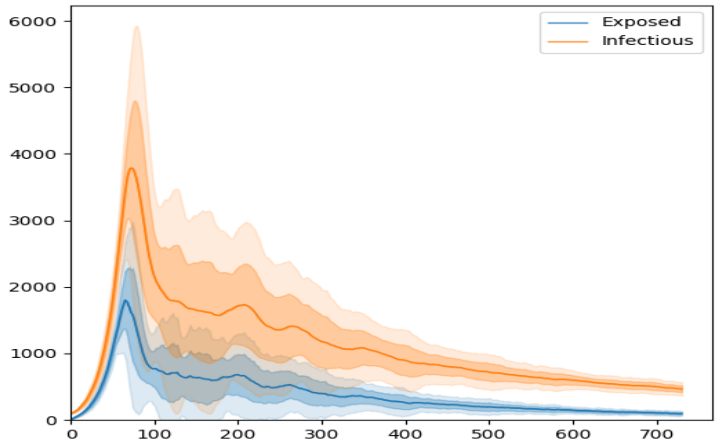
The voice of experience

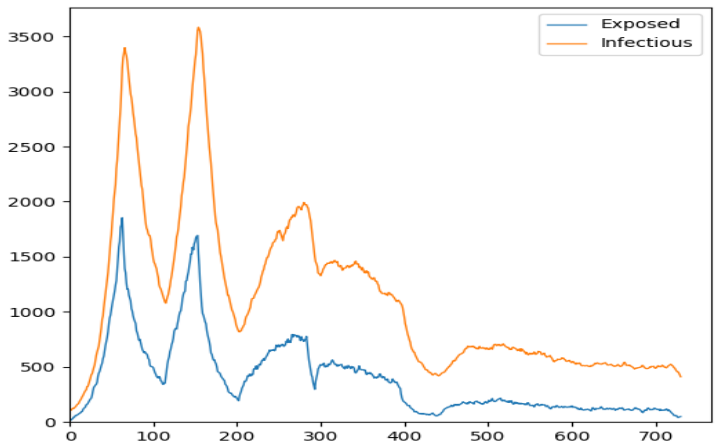


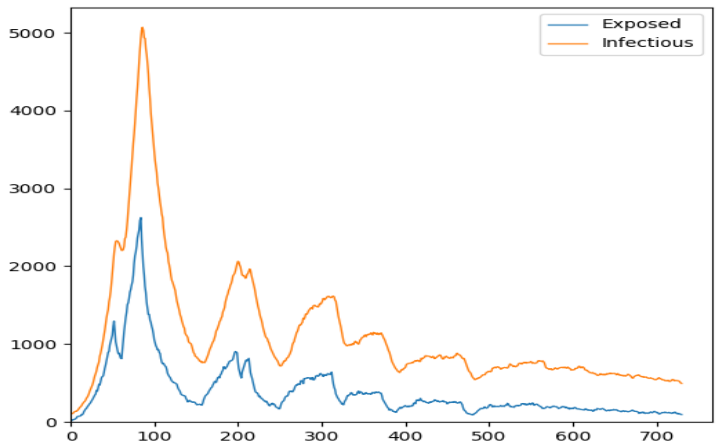
Trajectories and averages

Surveillance testing with triggers (FIND LMIC work)









William Waites, Matteo Cavaliere, David Manheim, Jasmina Panovska-Griffiths, and Vincent Danos. *Scaling up epidemiological models with rule-based modelling*. June 2020. arXiv: 2006.12077 [q-bio.PE]

This model:
`https://git.sr.ht/~wwaites/icms-modresp` (82 lines)

Kappa Language:
`https://kappalanguage.org`

Thank you

Funding: Chief Scientist Office COV/EDI/20/12

lfcs | Laboratory for Foundations of Computer Science

The University of Edinburgh