

Extensions to the pomp package and framework

Spencer J. Fox Qianying (Ruby) Lin Jesse Wheeler

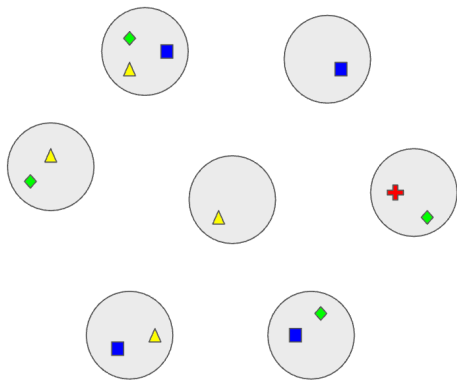
Package Extensions

Mathematically, POMP models are very versatile, and many of the ideas useful to the models described so far are also applicable in more general settings. This gives rise to a few useful extensions of the `pomp` package, which we try to describe here.

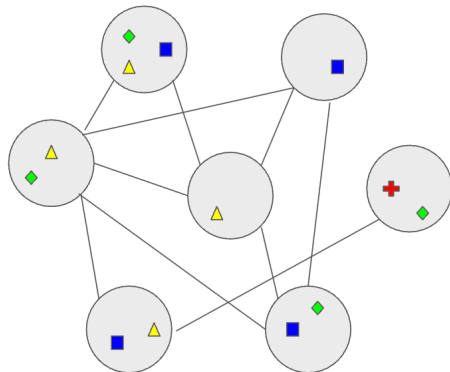
- ▶ Meta-population data and models
 - ▶ `panelPomp`
 - ▶ `spatPomp`
- ▶ Genomic / phylogenetic data
 - ▶ `phylopomp`

Each of these packages extend the `pomp` package to other useful cases.

Meta-population: Data and Models



panelPomp: data from related,
independent systems.



spatPomp: data from a single
interacting system.

Meta-population packages: Examples

Both packages require `pomp` to be installed, and the internal workings of both packages actually use many of the `pomp` functions. Thus, familiarity with `pomp` is a prerequisite.

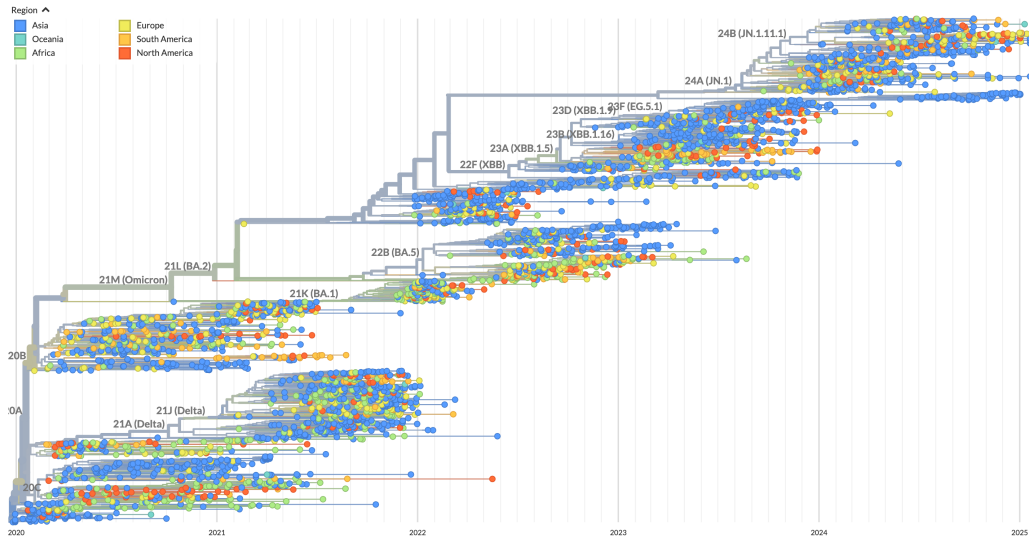
- ▶ `panelPomp`:

- ▶ We have *panel* or *longitudinal* data.
- ▶ Several independent yet related systems. Information from all systems is useful, not just one location.
- ▶ Simultaneously allows for features shared by each system as well as system unique features.
- ▶ Example: Late-stage COVID-19 measured in 3 locations: New York, London, Atlanta. Same disease in all places, but we can treat each as an (approximately) independent system. They may share features (recover rate, vaccine efficacy, etc.) that we want to estimate using all data, but may also have unique features at each place (reporting rate, contact rate, etc.)

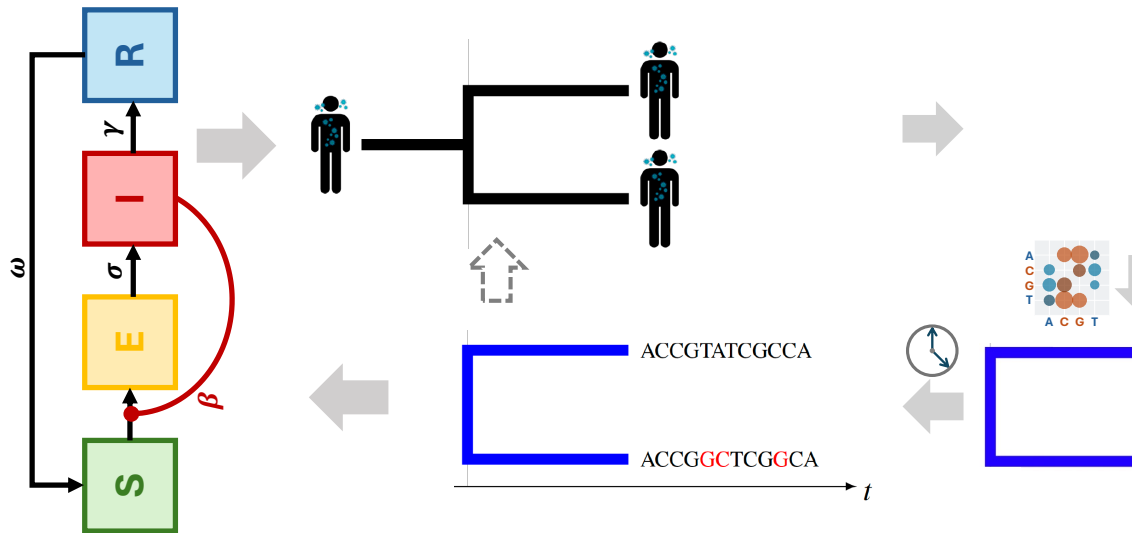
- ▶ `spatPomp`:

- ▶ The name comes from the idea that we have spatially explicit POMP models.
- ▶ The data at multiple locations are thought of as location-specific measurements from the same system. That is, the dynamic systems underlying the data are connected.
- ▶ Also allows for both shared and unique features for each location.

Background: genomic sequences & phylogenetic tree



Background: phylogenetics & phylodynamics



Installation

The github repo for phylopomp is on <https://github.com/kingaa/phylopomp/>. We can install the latest released version from github using devtools.

```
library(devtools)  
install_github("kingaa/phylopomp@0.14.8.0")
```

To confirm the package is installed successfully,

```
packageVersion("phylopomp")
```

```
[1] '0.14.9.1'
```

```
library(phylopomp)
```

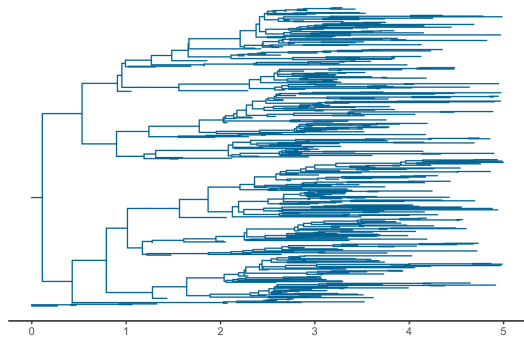
Simulation and inference

- ▶ The first useful function of `phylompomp` is to build customized models and simulate genealogies from it. Pre-defined models are included: linear birth-death model (`lbdp`), moran model, SIR, SEIR, two-class SIR model with super-infection (`si2r`), two-strain SIR model (`siir`), etc.
- ▶ The second is to infer the models given genealogies using `pomp`. This function is currently under development, while a few models are available: `lbdp`, `moran`, `sir`, and `seir`.

Example: simulation, pre-defined model I

```
set.seed(1234)
simulate("SIR",Beta=2,gamma=1,psi=2,S0=1000,I0=5,time=2) |>
  simulate(Beta=5,gamma=2,time=5,psi=3) -> model.sir # update params
plot(model.sir)
```

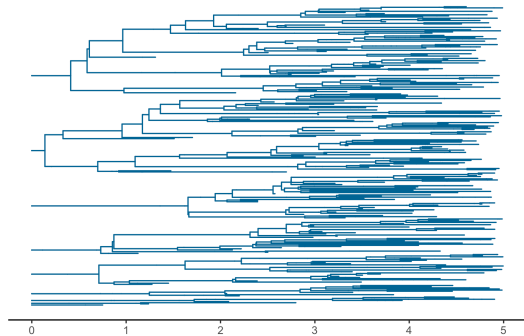
Example: simulation, pre-defined model II



Example: simulation, pre-defined model III

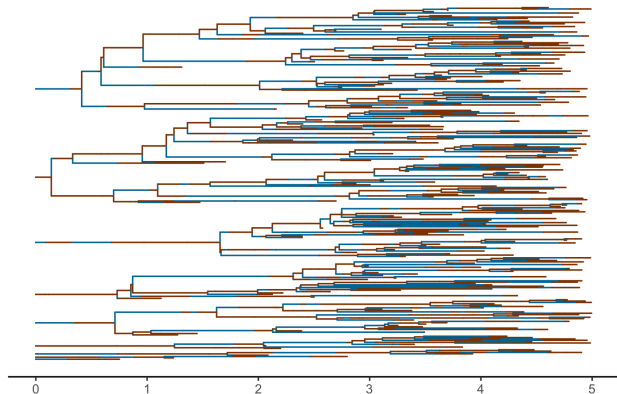
```
set.seed(1234)
simulate("SEIR",Beta=2,sigma=2,gamma=1,psi=2,S0=1000,I0=5,time=2) |>
  simulate(Beta=5,gamma=2,time=5,psi=3) -> model.seir
plot(model.seir)
```

Example: simulation, pre-defined model IV



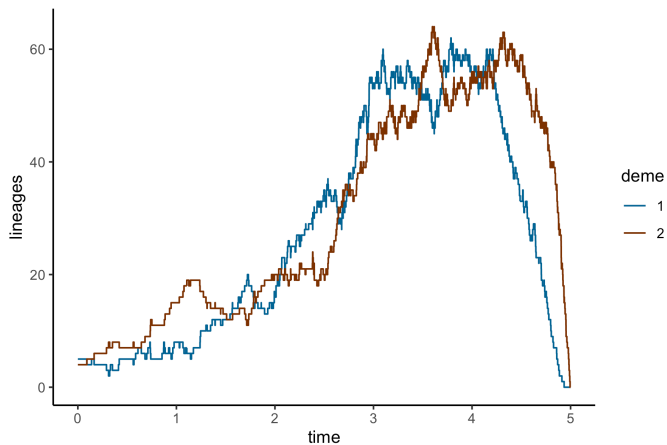
Example: simulation, pre-defined model V

```
plot(model.seir, obscure=FALSE)      # show E and I transitions
```



Example: simulation, pre-defined model VI

```
lineages(model.seir, obscure=FALSE) |> plot()
```



Example: inference, pre-defined model I

```
library(pomp)
set.seed(1234)
# simulate a geneal object from an SIR model
simulate("SIR",Beta=3,gamma=1,psi=2,omega=1,S0=100,I0=5,time=10) -> x
# build a pomp object
x |>
  sir_pomp(
    Beta=3,gamma=1,psi=2,omega=1,
    S0=100,I0=5,R0=0
  ) -> po
po |> pfilter(Np=5000) -> pf

pf |> logLik()
```

Example: inference, pre-defined model II

```
[1] -326.5652
```



Example: inference, pre-defined model III

```
set.seed(1234)
simulate("SEIRS",      # simulate a genealogy from an SEIR model
  Beta=4,sigma=1,gamma=1,psi=1,omega=1,
  S0=100,E0=3,I0=5,R0=100, time=5
) -> G
G |>
  seirs_pomp(
    Beta=4,sigma=1,gamma=1,psi=1,omega=1,
    S0=100,E0=3,I0=5,R0=100
  ) |> pfilter(Np=1000) |>
  replicate(n=20) |> concat() -> pf
pf |> logLik() |> logmeanexp(se=TRUE)
```

Example: inference, pre-defined model IV

est	se
-168.856637	1.265223

License, acknowledgments, and links

- ▶ This lesson is prepared for the Simulation-based Inference for Epidemiological Dynamics module at the Summer Institute in Statistics and Modeling in Infectious Diseases, SIS MID.
- ▶ The materials build on previous versions of this course and related courses.
- ▶ Licensed under the Creative Commons Attribution-NonCommercial license. Please share and remix non-commercially, mentioning its origin. 
- ▶ Produced with R version 4.5.1 and pomp version 6.3.
- ▶ Compiled on 2025-07-23.

[Back to Lesson](#)

[R code for this lesson](#)